A Principles and Practices Exam Specification to Support Software Engineering Licensure in the United States of America

PHILLIP A. LAPLANTE
Penn State

BETH KALINOWSKI
Prometric

MITCHELL THORNTON
Southern Methodist University

In April 2013 several states in the United States will require licensure for certain individuals who are involved in the creation of software that can affect the health, safety, and welfare of the public. It is expected that eventually, all states and jurisdictions in the United States will require such licensure. Each state has different licensure criteria, but all include certain educational and experiential requirements, passing two tests, with one being a common test of engineering fundamentals, and the other a test of minimal competency in relevant areas of software engineering knowledge and practice. While the common test of engineering fundamentals exists, the software engineering examination does not. In order to develop this examination, the authors conducted a study using a multimethod approach in identifying the professional activities and knowledge/skills that are important to the competent performance of software engineers who serve the public. In this article the authors describe the study, the results, and the test specification that was derived. Demographic information for the survey respondents is also presented.

Key words

professional licensure, professional knowledge studies, safety critical systems

INTRODUCTION

States license engineers and other professionals to ensure minimal competency in order to protect the public. In the case of engineers, licensure is mandatory for those offering their services directly to the public and not through an industrial or government entity (these engineers are exempted from licensure requirements in many jurisdictions). While the licensing of civil, mechanical, electrical, and other engineers has been mandatory in the United States for nearly 100 years, the case is not the same for software engineering. No licensing was required for software engineers until 1998 (Bagert 2002) when Texas began licensing those who participated in the development of systems that demonstrate “the application of mathematical, physical, or computer sciences to activities such as real-time and embedded systems, information or financial systems, user interfaces, and networks” (TBPE 2011). Texas discontinued licensing, however, in 2009 in anticipation of the development of a national examination.

In 2008, the IEEE-USA Licensure and Registration Committee, IEEE Computer Society (IEEE-CS), National Society of Professional Engineers (NSPE), and Texas Board of Professional Engineers (TBPE) formed the Software Engineering Licensure Consortium (SELC) and approached the NCEES, a nonprofit organization that assists state boards of professional engineering by developing licensure exams and assisting in developing licensure laws (Thornton and Laplante 2010) with a proposal to develop a software licensure exam. The SELC was able to show that: 1) software engineering was a distinct discipline of engineering; 2) at least 10 states would offer the exam if created; 3) there were many ABET-accredited software engineering baccalaureate programs in the United States; and 4) the SELC would coordinate the technical content expertise to produce the examination. Under these conditions, the NCEES agreed to support the creation and maintenance of a software engineering exam (Thornton and Laplante 2010).

When this examination is available in mid-2013, Alabama, Delaware, Florida, Michigan, Missouri, New Mexico, New York, North Carolina, and Virginia will join Texas in requiring licensure for software engineers involved in building software that could affect the health, safety, and welfare of the public. It is expected that the other states and U.S. jurisdictions will eventually follow suit (Laplante and Thornton 2010).

The purpose of this article is not to revisit the arguments for or against licensure of software engineers—the debate is well covered in the literature (for example, Frailey 1999; Knight and Leveson 2002; Moore 2003; Parnas 2002; Speed 1999; Kruchten 2008). Rather, the intent of this article is to establish confidence in the exam development methodology and to provide empirical data for use by software engineering researchers. The remainder of this article is as follows. First, the authors give the relevant history and context for the exam. Next, they describe the research methodology and survey population. Finally, they present and discuss the survey results and how they were used to establish the basis for the exam.

RESEARCH METHODOLOGY

A Professional Activities and Knowledge/Skills (PAKS) study refers to procedures designed to obtain descriptive information about the professional activities performed on a job and the knowledge, skills, or abilities thought necessary to adequately perform those professional activities. The PAKS study for the Principles and Practice of Engineering Software examination was designed to *The Standards for Educational and Psychological Testing (The Standards),* a comprehensive technical guide that provides criteria for the evaluation of tests and testing practices, and the effects of test use. *The Standards* was developed jointly by the American Psychological Association (APA), the American Educational Research Association (AERA), and the National Council on Measurement in Education (NCME) (APA 1999). The guidelines presented in *The Standards,* by professional consensus, have come to define the necessary components of quality testing. Consequently, a testing program that adheres to *The Standards* is more likely to be judged valid and defensible than one that does not.

> “The content domain to be covered by a credentialing test should be defined clearly and justified in terms of the importance of the content for credential-worthy performance in an occupation or profession. A rationale should be provided to support a claim that the knowledge/skills or skills being assessed are required for credential-worthy performance in an occupation and are consistent with the purpose for which the licensing or licensure program was instituted … Some form of job or PAKS study provides the primary basis for defining the content domain …” (APA 1999, 161)

A well-designed PAKS study includes the participation of a representative group of subject matter experts who reflect the diversity within the profession. Diversity refers to regional or job context factors and to individual factors such as experience, gender, and race/ethnicity. The demonstration of content validity is dependent on the judgments of subject matter experts. The process is enhanced by the inclusion of large numbers of subject matter experts who represent the diversity of the relevant areas of expertise.

Designing the Software Engineering PAKS Study

The NCEES convened a task force committee comprising a representative group of software professionals in September 2010. The purpose of the meeting was to develop the PAKS study content. The software engineering PAKS study consisted of several activities: survey development, survey dissemination, compilation of survey results, and test specifications development.

Activities conducted during the meeting included developing, reviewing, and revising the major domains, professional activities, and knowledge/skills that are necessary for competent performance by software engineers. The results of the Certified Software Development Professional Job Analysis were used to begin discussions. Survey rating scales and background and general information questions were presented, discussed, and revised.

Using the results of the task force meeting, a draft of the online survey was constructed. The following professional activities and knowledge/skills domains resulted:

1. Requirements
2. Design
3. Construction
4. Testing
5. Maintenance
6. Configuration management
7. Engineering processes
8. Quality assurance
9. Safety, security, and privacy

Each task force member received a copy of the draft survey in order to review their work and recommend any revisions. Comments provided by the task force committee for the online survey were compiled by the Prometric company staff and reviewed via a Web conference on October 18, 2010, with NCEES staff and the task force members. Refinements were incorporated, as recommended by the task force, into the online survey in preparation for a pilot test. Members of the task force nominated individuals for participation in the pilot test. The purpose of the small-scale pilot test was to have software professionals who had no previous involvement in the development of the survey review and offer suggestions for its improvement.

Twenty-two software professionals participated in the survey pilot test. Six respondents were licensed professional engineers, having taken the professional examination in areas other than software engineering. The pilot participants encompassed representatives from both the computer and noncomputer industry, eight states in the United States, and different levels of experience. Pilot participants were asked to review the survey for clarity of wording, ease of use, and comprehensiveness of content coverage. The survey was revised and finalized based on the review of the pilot test comments and administered on November 10.

The final version of the online survey consisted of four sections:

1: Background and General Information

2: Professional Activities

3: Knowledge/Skills

4: Recommendations for Test Content

In Section 1: Background and General Information, survey participants were asked to provide general and background information about themselves and their professional activities. In Section 2: Professional Activities and Section 3: Knowledge/Skills, survey participants were asked to rate the statements using the importance scales shown in Table 1.

Survey participants were asked to indicate how well the statements covered the professional activities and knowledge/skills within each domain. Respondents made their judgments using a five-point rating scale (1=very poorly; 2=poorly; 3=adequately;

TABLE 1 Importance scales for activities and knowledge/skills

Professional activities	Knowledge/Skills
Importance: How important is this professional activity for a newly licensed software engineer with four to six years of software engineering experience to practice in a manner that protects the health, safety, and welfare of the public?	Importance: How important is the knowledge/skill topic for a newly licensed software engineer with four to six years of software engineering experience to practice in a manner that protects the health, safety, and welfare of the public?
0 = Of no importance	0 = Of no importance
1 = Of little importance	1 = Of little importance
2 = Of moderate importance	2 = Of moderate importance
3 = Important	3 = Important
4 = Very important	4 = Very important

4=well; 5=very well). A write-in area was provided for respondents to note any areas that were not covered within a specific domain.

Survey participants were also asked to indicate the relative percentage the following knowledge/skills should receive on the software engineering examination.

1. Requirements
2. Design
3. Construction
4. Testing
5. Maintenance
6. Configuration management
7. Engineering processes
8. Quality assurance
9. Safety, security, and privacy

The derivation of test specifications from those statements verified as important by the surveyed professionals (based on a mean analysis of the importance ratings for professional activities and knowledge) provides a substantial evidential basis for the content validity (content relevance) of credentialing examinations.

Survey Population and Dissemination

The survey instrument was Web based and hosted on a secure third-party website. Invitees were determined from IEEE based on technical interest areas, NSPE, TBPE, and an IEEE-CS mailing list, as well as email addresses provided by the task force.

Previous Surveys of Software Knowledge Domain

Since the 1990s, the IEEE-CS and Association for Computing Machinery (ACM) have developed important infrastructure elements that help provide a basis for an appropriate examination of software engineering. For example, in the 1990s, the two organizations jointly developed a software engineering code of ethics. The code of ethics helps provide a basis for licensure as a means to protect the interests of the public: "Software engineers shall act in a manner that is in the best interests of their client and employer, consistent with the public interest" (Gotterbarn and Miller 2009).

In 2004, the joint ACM/IEEE-CS Computing Curriculum project published an undergraduate-level (bachelor's degree) software engineering curriculum paving the way for the establishment of ABET accredited programs in software engineering (Lethbridge et al. 2006). In the late 1990s, the IEEE-CS and nine cosponsors from various segments of industry and academe conducted a comprehensive study to characterize the scope of software engineering and describe its content. The resultant artifact, the *Guide to the Software Engineering Body of Knowledge* (SWEBOK), has been used as a reference model for a wide body of real software engineering projects and theoretical research (Bourque et al. 1999).

Concurrently, Carnegie Mellon University's Software Engineering Institute (SEI) established and refined its Capability Maturity Model (CMM) and its variants. The CMM is used to identify and model an organization's ability to produce software at a predictable cost and schedule. The CMM and its

TABLE 2 Knowledge/Skills statements by pass, borderline, and fail categories

Knowledge/Skills domains	No. of knowledge/ skills statements	Pass (mean 2.50 or above)	Borderline (mean 2.40 to 2.49)	Fail (mean less than 2.40)
1. Requirements	6	6	0	0
2. Design	8	8	0	0
3. Construction	3	3	0	0
4. Testing	5	5	0	0
5. Maintenance	3	3	0	0
6. Configuration management	6	3	2	1
7. Engineering processes	5	5	0	0
8. Quality assurance	3	3	0	0
9. Safety, security, and privacy	9	8	1	0
Total	**48**	**44**	**3**	**1**
Percentage	**100.00%**	**91.67%**	**6.25%**	**2.08%**

variants have been used by thousands of organizations worldwide, in all economic sectors, to improve their software development practices (Paulk 1995).

Starting in the mid 2000s, the IEEE-CS began developing its Certified Software Development Professional (CSDP) certificate program, which supports competency evaluation of software practitioners with four years of experience and a corresponding certification, the Certified Software Development Associate (CSDA) for those entering the field (Fernando Naveda and Seidman 2005). During the same timeframe, ASQ developed its Software Quality Engineer Certification and the associated body of knowledge (ASQ 2008).

More recently, representatives from academia, industry, government, and professional societies formed the integrated Software and Systems Engineering Curriculum project (iSSEc) to create a new reference curriculum, which involved a conceptual inventory of the software engineering discipline. The resulting artifact, the *Graduate Software Engineering Reference Curriculum (GSwERC),* provides guidelines and recommendations for master's level software engineering education (Pyster et al. 2009).

These efforts influenced the task force in the development of the PAKS study. These previous efforts, however, were quite broad and differed with respect to the software engineering discipline, and the PAKS study was narrow in that it was intended to validate the professional activities and knowledge/skills judged to be relevant to a newly licensed software engineer working on systems that affect the health, safety, and welfare of the public.

SURVEY RESULTS

Survey Response Rate

A total of 323 completed surveys were submitted via the online survey. Since this was the first PAKS study for professional software engineers, responses from licensed (N=64) and nonlicensed (N=259) engineers were included. The survey was emailed to an IEEE- and task force-provided list of software professionals containing 6,096 addresses, though, only 4,386 email addresses proved valid. There were 323 completed surveys for a rate of 7.36 percent. Based on the analysis of survey responses and subsequent conversation with the test specifications committee, it was deemed that a representative group of software professionals completed the survey in sufficient numbers to meet the requirements to conduct statistical analysis.

Demographic Characteristics of Survey Respondents

Demographic information for survey respondents is needed to ensure that the responses are valid across different genders, experience levels, geographic location, ethnicity, and so on. In many cases, comparisons

TABLE 3 Mean, standard deviation, and frequency distribution percentage of professional activities content coverage

	Content coverage						
			Frequency percentage				
Professional activities domain	**Mean**	**SD**	**1=Very poorly**	**2=Poorly**	**3=Adequately**	**4=Well**	**5=Very well**
1. Requirements	3.89	0.85	1.25%	2.49%	27.41%	43.93%	24.92%
2. Design	3.91	0.76	0.00%	2.85%	25.32%	50.00%	21.84%
3. Construction	3.83	0.78	0.00%	4.43%	26.58%	50.32%	18.67%
4. Testing	3.87	0.82	0.32%	4.10%	26.50%	46.06%	23.03%
5. Maintenance	3.82	0.81	0.32%	3.48%	30.70%	44.94%	20.57%
6. Configuration management	3.85	0.79	0.32%	3.21%	28.53%	47.12%	20.83%
7. Engineering processes	3.87	0.79	0.63%	2.53%	26.58%	49.37%	20.89%
8. Quality assurance	3.80	0.82	0.65%	2.94%	32.68%	42.81%	20.92%
9. Safety, security, and privacy	3.87	0.82	0.32%	2.85%	30.38%	42.41%	24.05%

TABLE 4 Mean, standard deviation, and frequency distribution percentage of knowledge/skills content coverage

	Content coverage						
			Frequency percentage				
Knowledge/Skills domain	**Mean**	**SD**	**1=Very poorly**	**2=Poorly**	**3=Adequately**	**4=Well**	**5=Very well**
1. Requirements	3.97	0.77	0.32%	0.97%	26.45%	45.81%	26.45%
2. Design	3.94	0.76	0.00%	1.63%	27.04%	46.58%	24.76%
3. Construction	3.82	0.86	0.65%	4.56%	29.97%	42.02%	22.80%
4. Testing	3.89	0.78	0.00%	2.30%	29.51%	44.92%	23.28%
5. Maintenance	3.79	0.79	0.00%	3.65%	32.89%	44.19%	19.27%
6. Configuration management	3.87	0.78	0.32%	1.62%	30.52%	45.45%	22.08%
7. Engineering processes	3.83	0.79	0.66%	2.30%	30.59%	46.38%	20.07%
8. Quality assurance	3.81	0.83	0.66%	2.98%	33.11%	41.39%	21.85%
9. Safety, security, and privacy	3.95	0.76	0.33%	0.65%	27.45%	47.06%	24.51%

between these subgroups could be made in order to ensure there was no disagreement between these groups. The survey participants represented a substantially varied group along many demographic dimensions. Details of the demographic characteristics of participants can be found in the online supplement to this article.

Professional Activities and Knowledge Areas

Of the 186 professional activities and knowledge/skills, 152 (81.72 percent) achieved high means (at or above 2.50), thereby validating their importance to competent performance for software engineers.

Of the 48 knowledge/skills, 44 (91.67 percent) achieved high importance means. Table 2 shows the professional activities that were placed in pass, borderline, and fail categories by domain.

Subgroup Analysis of Professional Activities and Knowledge/ Skills Ratings

The index of agreement (IOA) is a measure of the extent to which subgroups of respondents agree on which professional activities and knowledge/skills are important (Cohen 1968). The index of agreement provides a method of computing the similarity in judgments between groups that is more tailored to the purpose of a PAKS study than the correlation coefficient. Although the correlation coefficient measures the tendency toward agreement along the full range of possible ratings, the agreement index focuses on whether two groups agree that the content should (or should not) be included in an examination. The exam IOA for exam content was very high along most major subgroups including gender, years of experience, and geographic location. Details of the IOA analysis can be found in the online supplement to this article.

TABLE 5 Survey respondents' test content recommendations by mean percentages and standard deviations

Topic areas	Mean (%)	SD (%)	Range	
			Minimum	Maximum
1. Requirements	14.37	5.49	0	40
2. Design	15.91	5.55	1	50
3. Construction	13.64	6.44	0	40
4. Testing	13.62	4.74	4	30
5. Maintenance	8.13	3.63	0	20
6. Configuration management	7.14	3.58	0	20
7. Engineering processes	9.47	4.57	0	25
8. Quality assurance	7.96	4.05	0	20
9. Safety, security, and privacy	9.75	4.98	0	30

TABLE 6 Test content weights recommended by the test specifications committee

Knowledge/Skills domains	No. of knowledge/skills statements	TS committee recommended percentages	Number of examination items
1. Requirements	6	17%	14
2. Design	8	14%	11
3. Construction	3	11%	9
4. Testing	5	12%	10
5. Maintenance	3	8%	6
6. Configuration management	4	8%	6
7. Engineering processes	5	8%	6
8. Quality assurance	3	8%	6
9. Safety, security, and privacy	8	14%	12
Total	**45**	**100.00%**	**80**

Content Coverage Ratings

The survey participants were asked to indicate how well the statements within each of the professional activities and knowledge/skills domains covered important aspects of that area. These responses provide an indication of the adequacy (comprehensiveness) of the survey content (see Tables 3 and 4).

The five-point rating scale included 1=very poorly, 2=poorly, 3=adequately, 4=well, and 5=very well. The means and standard deviations for the professional activities and knowledge/skills ratings are provided in Tables 4 and 5. For the professional activities domains, the means ranged from 3.80 to 3.91. The means across the knowledge/skills domains ranged from 3.79 to 3.97. These means provide evidence that the professional activities and knowledge/skills were adequately to well covered on the survey.

Survey respondents were also asked to write in professional activities or knowledge/skills that they believe should be included in the listing of important professional activities and knowledge/skills.

Test Content Recommendations

In survey Section 4: Recommendations for Test Content, participants were asked to weigh each domain out of 100 possible items for a professional software engineering examination. This information was used by the test specifications committee as an aid in making decisions about how much emphasis the knowledge/skills domains should receive on the test content outline. The mean weights across all survey respondents are presented in Table 5.

DEVELOPMENT OF THE TEST SPECIFICATIONS

The exam development committee participated in a wideband Delphi activity that required each member to individually assign a percentage weight to each of the knowledge/skill domains. Weights were then entered into a spreadsheet and shown to the committee. The committee members were able to compare the test content weights derived from the survey responses to their own estimates. This resulted in a discussion among the committee members regarding the optimal percentages for the software examination. Table 6 presents the test specifications recommendations, including the percentage content weights by domain and the target number of questions for the examination.

Professional Activities and Knowledge/Skills Included on the Test Specifications

The test specifications committee reviewed the professional activities and knowledge/skills results to make final recommendations about the content (domain) areas that should be included on the examination.

The survey results served as the primary source of information used by the test specifications committee to make test content decisions. Recommendations were based on the following criteria:

- The mean importance ratings
- The frequency distribution of importance ratings
- The mean importance ratings for specific subgroups (including years in software engineering, job function, role, and type of software produced)
- The appropriateness of the content for the examination

Based on information obtained from the survey, data analyses by respondent subgroups (for example, job title) are possible when sample size permits. A subgroup category is required to have at least 30 respondents to be included in the mean analyses. This is a necessary condition to ensure that the mean value based upon the sample of respondents is an accurate estimate of the corresponding population mean value.

The following quantitative data analyses were produced:

- Means, standard deviations, and frequency (percentage) distributions for professional activities and knowledge/skills importance and content coverage ratings
- Means and standard deviations for test content recommendations
- Index of agreement values for designated subgroups

Since a major purpose of the survey was to ensure that only important professional activities and knowledge/skills are included in the development of test specifications, a criterion (cut point) for inclusion needs to be established.

A criterion that has been used in similar studies is a mean importance rating that represents the midpoint between moderately important and important. For the importance-rating scale used across many studies, the following judgment rubric was used:

	Means
Pass:	At or above 2.50
Borderline:	2.40 to 2.49
Fail:	Less than 2.40

This criterion is consistent with the intent of content validity, which is to measure only important knowledge/skills in the credentialing examination. For this PAKS study, the value of this criterion was set at 2.50, which is the same as is used in P&P examinations for other engineering disciplines.

The professional activities and knowledge/skills were placed into one of three categories—pass, borderline, or fail—based on their mean importance ratings:

TABLE 7 Professional activities edited/included on the test specifications by the test specifications committee

Domain	Professional activity	Included on the test specification?	Rationale
Maintenance	Perform perfective maintenance	Yes	Important to software architects (2.59); more than 84% thought it was at least moderately important
Configuration management	Establish tool sets and software libraries (e.g., compilers, build scripts)	Yes	Modified for clarity; more than 50% thought it was either important or very important
Engineering processes	Define process and product metrics for project	Yes	More than 51% of respondents thought it was either important or very important

TABLE 8 Knowledge/Skills included on the test specifications by the test specifications committee

Domain	Knowledge/Skill	Included on the test specification?	Rationale
Configuration management	Configuration status accounting (for example, configuration status information, configuration status reporting)	Yes	Important to software or systems analyst; more than 86% of respondents thought it was at least of moderate importance

- The pass category contains those statements whose mean ratings are at or above 2.50, and are considered eligible for inclusion in the development of test specifications.
- The borderline category contains those statements whose mean ratings are between 2.40 and 2.49. The borderline category is included to provide a point of discussion for the test specifications committee to determine if the statement warrants inclusion in the test specifications.
- The fail category contains those statements whose mean ratings are less than 2.40. It is recommended that statements in the fail category be excluded from consideration in the test specifications.

If the test specifications committee felt that a statement rated below 2.50 should be included in the specifications and can provide compelling written rationale, those statements may be considered for inclusion. For example, although a professional activity or knowledge/skill may have a mean rating of less than 2.50, more than 50 percent of the respondents may have rated the statement as important or very important. In this instance, the test specifications committee might recommend the inclusion of the statement on the test specifications. The written rationale would note that a majority of the survey respondents rated the statement as important.

The test specifications committee recommended the inclusion of 36 professional activities and 55 knowledge/skills.

Professional Activities Recommended for Inclusion

A total of 108 of the 138 professional activities achieved mean ratings at or above 2.50 for the overall group (pass category) and were included on the test specifications. Twelve statements achieved mean ratings less than 2.50 but greater than 2.40 (borderline category). Three of the statements were approved for use in the test specifications (see Table 7). Eighteen statements achieved mean ratings less than 2.40 (fail category). None of the statements were approved for use in the test specifications.

Knowledge/Skills Recommended for Inclusion

A total of 44 of the 48 knowledge/skills achieved mean ratings at or above 2.50 for the overall group

TABLE 9 Knowledge/Skill linkages to professional activities for requirements knowledge area

Knowledge/Skill (Bold) *Professional Activity (Italics)*
1. Software requirements fundamentals (for example, concept of operations, types of requirements, product and process requirements, functional and nonfunctional requirements, quantifiable requirements, system requirements, software requirements, derived requirements, constraints, service level)
15. Develop concept of operations
5. Define stakeholder requirements
10. Specify derived requirements
2. Requirements elicitation (for example, requirements sources, elicitation techniques, requirements representation)
3. Conduct elicitations
1. Identify stakeholders
3. Requirements specification (for example, System Definition Document, System/Subsystems Specification, Software Requirements Specification, Interface Requirements Specification)
14. Identify interface requirements
13. Draft requirements specifications
6. Develop derived safety, security, and privacy requirements
4. Requirements analysis (for example, requirements classification, conceptual modeling, architectural design and requirements, requirements allocation, requirements negotiation, formal methods, feasibility analysis)
6. Conduct feasibility studies
4. Conduct requirements analysis
9. Draft top-level architecture
7. Negotiate requirements
16. Allocate requirements to top-level architecture
5. Requirements verification and validation (for example, requirements reviews, prototyping, model validation, simulation)
11. Validate requirements
8. Verify requirements
6. Requirements management (for example, iterative nature of the requirements process, change management, requirement attributes, requirements traceability, measuring requirements, software requirements tools)
17. Manage changes to the requirements
7. Negotiate requirements
11. Validate requirements

(pass category) and were approved for inclusion on the test specifications. Six of the passing statements were modified for clarity. Three statements achieved mean ratings less than 2.50 but greater than 2.40 (borderline category). One of these borderline statements was approved for use in the test specifications (see Table 8).

Linking

Linking professional activities and knowledge/skills verifies that each knowledge/skill area included on an examination is related to the competent performance of important professional activities. As such, linking documents the content validity of the professional activities included in the test specifications.

Linking does not require the production of an exhaustive listing; rather, professional activity-knowledge/skill links are developed to ensure that each knowledge is identified as being related to the performance of at least one, or in most cases several, important professional activities.

Linking also provides guidance for item-writing activities. When item writers develop questions for specific professional activity domains, they have a listing of knowledge/skills that relate to the professional activities. This provides context for developing examination questions, and assists the item writers in question design. For example, linked knowledge areas and professional activities for requirements engineering are shown in Table 9.

THREATS TO VALIDITY

As is the nature of survey-based research, the validity of results can be threatened by the composition of the survey pool. The sample population for this research was mostly drawn from IEEE-CS and IEEE USA lists. It is unknown if a different set of results would have been obtained by surveying a list of self-identified software practitioners from other professional organizations, for example, the ACM.

Furthermore, survey respondents included licensed professional engineers, unlicensed software engineering practitioners, and engineers licensed in other countries. It is not clear if there are any significant differences in the views of these respective groups, though analysis, using the indices of agreement presented in the previous section, suggests that there are none.

Another threat concerns the possible different interpretations of the meaning of the statement "in a manner that protects the health, safety, and welfare of the public," which represented the cue that focused the responses on truly critical systems. Different individuals will likely assess the potential effects of failure for different systems in different ways, possibly biasing the interpretation of the kinds of systems that would come under the jurisdiction of licensure laws. Indeed, except in the state of Texas, these laws have yet to be written.

Finally, there is controversy as to the need for professional licensure and it is possible that those who disagreed with the need for licensure opted out of the survey upon receiving an invitation, thus biasing the results somehow.

CONCLUSIONS

The PAKS study for the Principles and Practices of Software Engineering examination was conducted to identify and validate professional activities and knowledge/skills important to the work performed by software engineers. Further, it was used to create test specifications that may be used to develop the Principles and Practices of Software Engineering examination. Although intended to aid in the development of the examination, these data and the resulting analysis should prove to be useful for researchers in other areas of software engineering professional activities.

The professional activities and knowledge/skills were developed through an iterative process involving the combined efforts of NCEES, subject matter experts, and Prometric company staff. These statements were then entered into a survey format and subjected to verification/refutation through the dissemination of a survey to software professionals. The survey participants were asked to rate the importance of specific professional activities and knowledge/skills for a software engineer to practice in a manner that protects the health, safety, and welfare of the public.

The results of the study support the following:

- All of the professional activities and knowledge/skills that were verified as important through the survey provide the foundation of empirically derived information from which to develop test specifications for the PE Software Engineering examination.
- Evidence was provided in this study that the comprehensiveness of the content within the professional activities and knowledge/skill domains was adequately to well covered.

The study used a multimethod approach to identify the professional activities and knowledge/skills important to the work performed by software engineers.

The software engineering exam is consistent with the Principles and Practices exam for other engineering disciplines. That is, it is an 80-question, eight-hour, open-book, multiple-choice exam. The first Principles and Practices examination will be offered in at least 10 states in April 2013 and annually thereafter. No one knows, however, how many individuals will take this exam, nor does anyone know how many engineers will eventually come under the jurisdiction of software engineering licensure laws in the United States.

REFERENCES

American Society for Quality (ASQ). 2008. *Software Quality Engineer Certification Body of Knowledge*. 1999. Available at: http://prdweb.asq.org/certification/control/software-quality-engineer/bok.

APA. 1999. *The Standards for Educational and Psychological Testing.* Washington, DC: American Psychological Association, American Educational Research Association, National Council on Measurement in Education.

Bagert, D. J. 2002. Texas licensing of software engineers: All's quiet, for now. *Communications of the ACM* 45, no. 11 (November): 92-95.

Bourque, P., R. Dupuis, A. Abran, J. W. Moore, and L. Tripp. 1999. The guide to the software engineering body of knowledge. *IEEE Software* (November/December): 35-44.

Cohen, J. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin* 70, no. 4 (October): 213-220.

Frailey, D. J. 1999. Software engineering grows up. *IEEE Software* (Nov/Dec): 66, 69.

Gotterbarn, D., and K. W. Miller. 2009. The public is the priority: Making decisions using the Software Engineering Code of Ethics. *Computer* (June): 66-73.

Knight, J. C., and N. G. Leveson. 2002. Should software engineers be licensed? *Commun. ACM* 45, no. 11 (November): 87-90.

Kruchten, P. 2008. Licensing software engineers? *Software* 25, no. 6 (Nov/Dec): 35-37.

Laplante, P., and M. Thornton. 2010. Licensing professional software engineers in the United States of America. In *Encyclopedia of Software Engineering.* Boca Raton: Auerbach Publications.

Lethbridge, T. C., R. J. LeBlanc, A. E. K. Sobel, T. B Hilburn, and J. L. Diaz-Herrera. 2006. SE2004: Recommendations for undergraduate software engineering curricula. *IEEE Software* (November/December): 19-25.

Moore, M. M. 2003. A license to practice software engineering. *IEEE Software* 20, no. 3 (May-June): 112-113.

Naveda, J. F., and S. B. Seidman. 2005. Professional certification of software engineers: The CSDP Program. *IEEE Software* (September/October): 73-77.

NCEES. 2011. NCEES vision, mission and strategic plan. Available at: http://www.ncees.org/About_NCEES/Vision_mission_and_strategic_plan.php.

Parnas, D. L. 2002. Texas licensing of software engineers: All's quiet, for now. *Commun. ACM* 45, no. 11 (November): 96-98.

Paulk, M. C. 1995. How ISO 9001 compares with the CMM. *IEEE Software* (January): 74-83.

Pyster, A., and R. Turner, H. Devanandham, K. Lasfer, and L. Bernstein. 2009. Master's degrees in software engineering: An analysis of 28 university programs. *IEEE Software* (September/October): 94-101.

Speed, J. R. 1999. What do you mean I can't call myself a software engineer? *IEEE Software* 16, no. 6 (Nov/Dec): 45-50.

TBPE. 2011. Statute, Texas Board of Professional Engineers. Available at: http://www.tbpe.state.tx.us/.

Thornton, M., and P. Laplante. 2010. Update on software engineering licensure initiative. *Today's Engineer.* Reprinted in *IEEE-USA in ACTION* (December 10): 7-8.

BIOGRAPHIES

Phil Laplante is professor of software engineering at The Pennsylvania State University. He received his bachelor's, master's, and doctorate degrees from Stevens Institute of Technology, and an MBA from the University of Colorado. Prior to his academic posts, he spent many years as a software engineer working on avionics, CAD, and telecommunications systems, including time at the Bell Labs Software Quality Assurance Center. His research interests are in software project management, requirements engineering, and software testing. Laplante chairs the committee that develops and maintains the licensure examination for software engineers in the United States. He can be reached by email at plaplante@psu.edu.

Beth Kalinowski is a test developer for Prometric, specializing in conducting job analysis research. In particular, she is responsible for the production of job analysis surveys and performing analysis of the survey data using SPSS. She has conducted many job analyses for licensure and certification programs across many professions including architecture, financial services, healthcare, and engineering. Before joining Prometric, Kalinowski worked as an office supervisor for a nonprofit organization and was a classroom teacher with Teach for America. She earned an MBA in 2006, and also holds a bachelor's degree in English and history from Bates College in Lewiston, Maine. She can be contacted by email at Beth.Kalinowski@prometric.com.

Mitchell A. Thornton is a professor of computer science and engineering at Southern Methodist University (SMU). Prior to his appointment at SMU, he served on the faculties of Mississippi State University and the University of Arkansas. Before entering academia, Thornton was employed by E-Systems Inc. for five years and Cyrix Corp. for one year. He has been active in professional licensure issues for many years and has served as chair of the NCEES committee for PE examination preparation for electrical and computer engineers and as chair of the IEEE-USA licensure and registration committee. Thornton is a professional engineer in Texas, Mississippi, and Arkansas, and is a Senior member of the IEEE and the ACM. He can be reached by email at mitch@lyle.smu.edu.