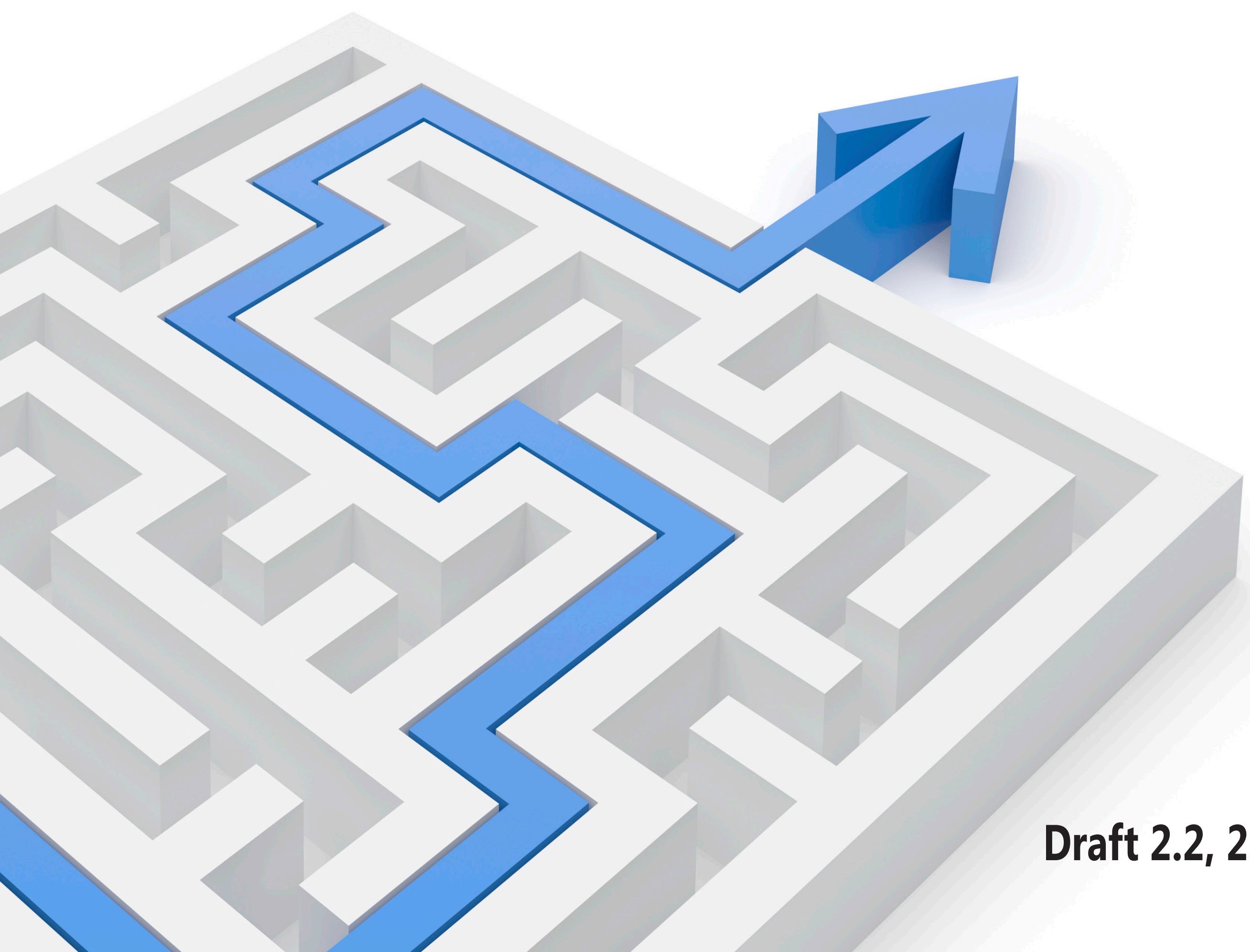
IEEE USA

SOFTWARE ENGINEERING PRACTICE STANDARDS
Licensure Guide



Draft 2.2, 22 June 2012

Table of Contents

List of Contributors

From IEEE-USA

- Don Bagert, CSDP, PE, PhD
- Phillip Laplante, CSDP, PE, PhD
- Mitchell Thornton, PhD, PE
- David Whitman, PhD, PE
- Gregg Vaughn, PhD, PE

From Texas Board of Professional Engineers

- David Howell, PE
- Lance Kinney, PE

From IEEE Computer Society

- James W. Moore, CSDP
- Pieter Botman, P.Eng.
- Paul Croll
- Richard Fairley, PhD
- Roger Fujii
- Stephen Georgas, Esq, P.Eng.
- Katherine Guillemette, IEEE Computer Society staff
- Paul Joannou, P.Eng.
- Susan K. Land, CSDP
- Timothy Lethbridge, PhD, P.Eng., ISP, CSDP
- Fraser Mann, Esq
- Chris Ruoff, IEEE Computer Society staff

Reviewed by

- Don R. Gilman, PE, Texas SWE Taskforce
- Bill Carroll, PhD, PE, Texas SWE Taskforce
- David Hensley, PE, Texas SWE Taskforce
- Dennis Frailey, PhD, Texas SWE Taskforce
- Steve Tockey, Texas SWE Taskforce

Team Genesis & Development

The team was created as a result of an action item at the Software Engineering Licensure Committee (SELC) meeting, a committee comprised of representatives from the IEEE USA, National Society of Professional Engineers, Texas Board of Professional Engineers, and IEEE Computer Society. It was mentioned that state licensing boards might welcome guidance in dealing with software engineering. Lance Kinney, Executive Director for TBPE, and Jim Moore, Vice President of Professional Activities for IEEE Computer Society, agreed to draft a paper. Subsequently, David Howell, Director of Licensing for TPBE, and Jim Moore agreed to co-chair an ad hoc group to create a draft. Jim recruited volunteers with relevant experience from the Computer Society. David and Lance solicited suggestions from other licensing boards.

The ad hoc group conducted its first and only meeting on 12 November 2010. Subsequent deliberation was performed via email and web/telecom meetings.

An initial mash-up was performed based on documents from various sources dealing with the issues of licensing software engineers. This document proved to be too large and too diffuse for its intended audience, licensing boards.

A second, much shorter document was drafted that was organized according to sections of the Model Law. The document provides guidance for applying each section of the Model Law when considering the characteristics of software engineering.

This draft was refined several times. It was presented to the 15 July 2011 meeting of the Software Engineering Licensing Consortium for their use in developing a document representing a broader consensus.

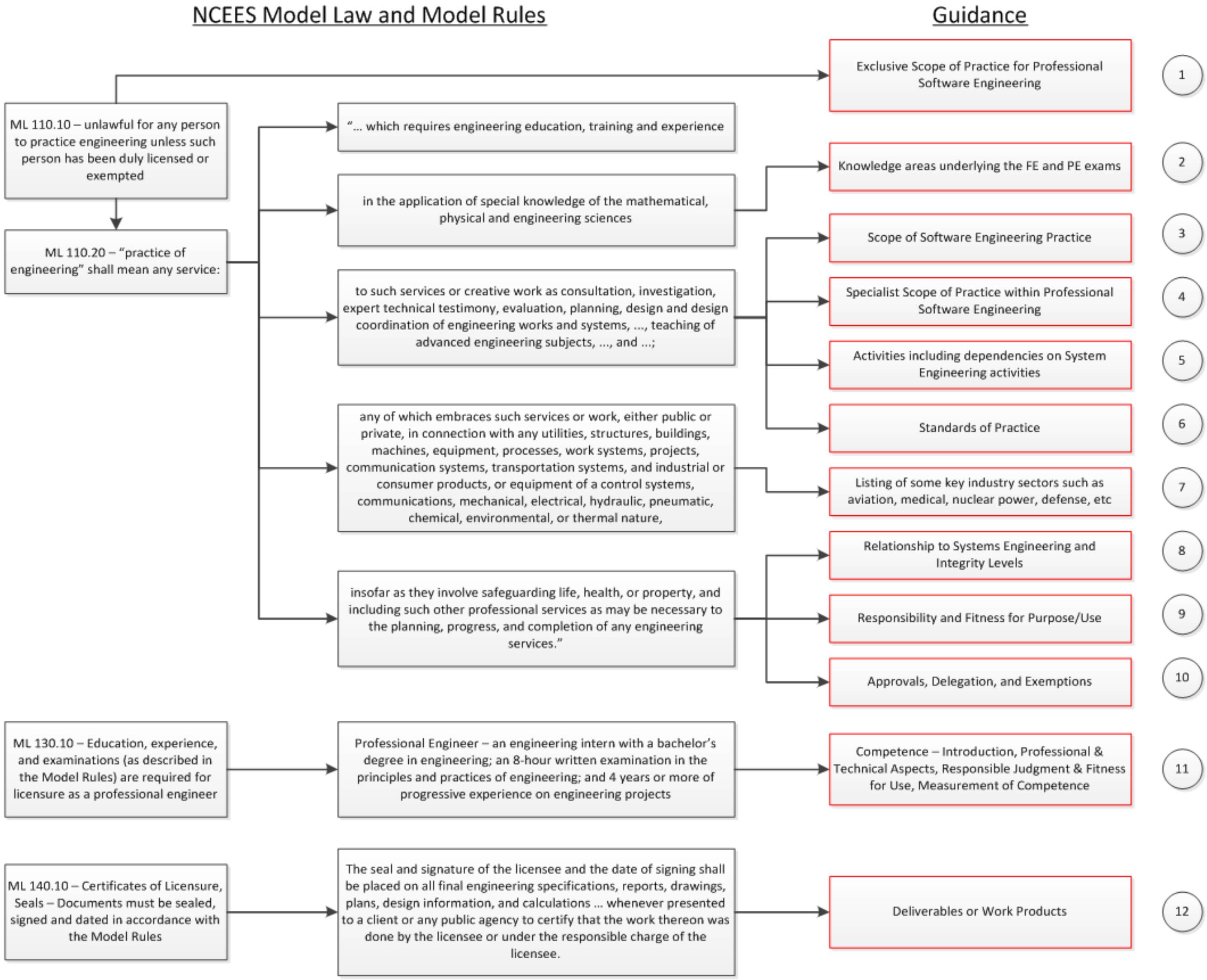
The document produced on 15 July 2011 was reviewed, edited, refined, and updated by members of the IEEE-USA Licensure and Registration Committee (LRC) as a guide for state licensing boards. The LRC is responsible for monitoring and contributing to USA professional engineering licensure on behalf of its constituents, the members of IEEE. IEEE-USA also serves as the official lead technical organization responsible for subject matter content in the NCEES Principles and Practices (PE) examination for professional licensure of engineers who practice in the area of software engineering.

The revised document was sent from the LRC to Lance Kinney in December 2011. The document was distributed to members of the Texas Software Engineering Taskforce for review resulting in this version.

National Council of Examiners for Engineers and Surveyors (NCEES) *Model Law* and *Model Rules*

Figure 1 shows some relevant sections of the NCEES *Model Law* and *Model Rules* that can be found at http://ncees.org/About_NCEES/Publications.php and some general guidelines for specific items that would pertain to the practice of Software Engineering and these documents. Each of the numbered items will form the basis for a section of this guide.

Figure 1. *Model Law* and *Model Rules* and Guidance



Section 1 - Exclusive Scope of Practice for Professional Software Engineering

Relevant Section: *Model Law* 110.10 General Provisions

> **A. Regulation of Engineers and Surveyors**—In order to safeguard life, health, and property and to promote the public welfare, the practice of engineering and/or the practice of surveying in this jurisdiction is hereby declared to be subject to regulation in the public interest. It shall be unlawful for any person to practice, or to offer to practice, engineering and/or surveying in this jurisdiction, as defined in the provisions of this Act, or to use in connection with their name or otherwise assume, or advertise any title or description tending to convey the impression that they are a licensed engineer and/or surveyor, unless such person has been duly licensed or authorized or is exempted under the provisions of this Act. The practice of engineering or surveying shall be deemed a privilege granted by this jurisdiction through the licensing board based on the qualifications of the individual as evidenced by their certificate of licensure, which shall not be transferable.

Guidance:

Software has become a pervasive component of many systems, devices and tools. In cases where the failure of the software can lead to a risk to life, health, or property then the development of the software falls under the practice of engineering and must be performed by a licensed engineer with knowledge in the area of software.

Software failures can result in safety risks, security risks, economic risks, or risks in other risk dimensions. It should be noted that for software such risks often occur in application domains not usually associated with engineering e.g. financial systems, and online databases. The software component's role within a system determines the relationship between the software component and system risks.

Standard ISO/IEC 15026:1998 defines the concept of system and software integrity levels. Integrity levels correspond to the level of risk associated with the system (or software component). The standard defines the concept of integrity levels but does not prescribe a set of levels; this is typically provided by industry-specific standards.

In order to safeguard life, health, and property and to promote the public welfare, the development of software having a high risk (integrity level) requires the services of licensed engineers with knowledge in the area of software. Software of lower risk (integrity level) does not; however, it shall ultimately be the responsibility of the licensing boards to determine where this risk differentiation lies within their jurisdictions.

Section 2 - Knowledge Areas Underlying the FE and PE Exams

Relevant Section: *Model Law* 110.20, Definitions; A. Engineer; 5. Practice of Engineering; paragraph 1;

> Practice of Engineering—The term "Practice of Engineering," as used in this Act, shall mean any service or creative work, the adequate performance of which requires engineering education, training, and experience in the application of special knowledge of the mathematical, physical, and engineering sciences to such services or creative work as …

Guidance:

Examinations are key elements of the assessment of competency for licensing engineers of any discipline. State licensing Boards use the NCEES Fundamentals of Engineering (FE) exam and the Principles and Practice of Engineering (PE) exam to assess an applicant's grasp of the "Special Knowledge of the mathematical, physical, and engineering sciences". Successful completion of the examinations provides evidence of minimum competency in the discipline examined.

The FE exam is available to engineering students and graduates of all disciplines and different versions of the examination are available for various engineering disciplines. It tests general knowledge of common subject areas as well as the more popular discipline specific areas. Software engineers would be expected to be able to successfully "pass" the FE exam as required for any other engineering discipline. The discipline specific section of the examination can be taken in one of the following areas:

- Chemical
- Civil
- Electrical, Computer, and Software
- Environmental
- Industrial
- Mechanical
- Other Disciplines

It is noted that the name "Electrical, Computer, and Software" is updated from the formerly named "Electrical" FE examination as of January 2013. This new name reflects the significant sections on software and computer engineering that are anticipated in the new examination. The authors of this guide strongly encourage prospective software engineering licensees to sit for the "Electrical, Computer, and Software" FE examination as it contains the most appropriate subject matter content for the discipline of software engineering.

The PE exam is designed to test more specific knowledge related to an engineer's field of practice. Minimally competent software engineers would be expected to be as able to successfully "pass" the PE exam designed for the software engineering discipline. The software engineering PE exam will assess a potential licensee's competency in the following areas of software engineering:

1. Requirements
2. Design
3. Construction
4. Testing
5. Maintenance
6. Configuration Management
7. Engineering Processes
8. Quality Assurance
9. Safety, Security, and Privacy

These areas were determined by an extensive Professional Activities and Knowledge/Skills (PAKS) study conducted by the NCEES in cooperation with the SELC.

The software engineering PE exam is being developed and will be maintained by NCEES, who will enlist subject matter experts from IEEE and other professional organizations, as appropriate. The questions on the exam are developed by practicing software engineers who have significant expertise in one or more of the subject matter areas listed above and in application domains where software can directly affect the health, safety and welfare of the public such as transportation, power generation, and medical systems.

Section 3 - The Scope of Software Engineering Practice

Relevant Section: *Model Law* 110.20, Definitions; A. Engineer; 5. Practice of Engineering; paragraph 1;

> **Practice of Engineering**—The term "Practice of Engineering," as used in this Act, shall mean any service or creative work, the adequate performance of which requires engineering education, training, and experience in the application of special knowledge of the mathematical, physical, and engineering sciences to such services or creative work as consultation, investigation, expert technical testimony, evaluation, planning, design and design coordination of engineering works and systems, planning the use of land, air, and water, teaching of advanced engineering subjects, performing engineering surveys and studies, and the review and/or management of construction for the purpose of monitoring and/or ensuring compliance with drawings and specifications; any of which embraces such services or work, either public or private, in connection with any utilities, structures, buildings, machines, equipment, processes, work systems, projects, communication systems, transportation systems, and industrial or consumer products, or equipment of a control systems, communications, mechanical, electrical, hydraulic, pneumatic, chemical, environmental, or thermal nature, insofar as they involve safeguarding life, health, or property, and including such other professional services as may be necessary to the planning, progress, and completion of any engineering services.

Guidance:

There are many sources that define the scope of software engineering practice. Several universities have EAC-ABET accredited programs in software engineering and they use a wide variety of textbooks and other curricular sources. In terms of professional practice, various technical societies have codes of conduct and ethics regarding professional software engineering practice.

The IEEE Computer Society has produced The Guide to the Software Engineering Body of Knowledge (SWEBOK), available at http://www.computer.org/portal/web/swebok. The SWEBOK Guide specifies the software engineering knowledge and related areas. Skills in applying this knowledge are obtained by experience in one or more application domains. An experienced software engineer should thus possess knowledge and skills in the various SWEBOK knowledge areas, as applied to one or more application domains in the context of safety critical systems or those that otherwise directly affect the public welfare.

The SWEBOK is intended to be a definitive statement of the scope of activities in which software engineers are engaged. SWEBOK can be used to provide the foundation for curricula, training programs, and certifications in software engineering. This volume can serve as a useful resource for determination of the scope of professional software engineering practice.

The SWEBOK, however, is broad in scope and covers the entire domain of software engineering. The nine knowledge areas determined from the PAKS study and listed above are contained in the SWEBOK, and have been determined through that study to be pertinent to software engineers working specifically on those systems that affect the health, safety and welfare of the public.

Section 4 - Specialist Scope of Practice within Software Engineering

Relevant Section: *Model Law* 110.20, Definitions; A. Engineer; 5. Practice of Engineering; paragraph 1;

> **Practice of Engineering**—The term "Practice of Engineering," as used in this Act, shall mean any service or creative work, the adequate performance of which requires engineering education, training, and experience in the application of special knowledge of the mathematical, physical, and engineering sciences to such services or creative work as consultation, investigation, expert technical testimony, evaluation, planning, design and design coordination of engineering works and systems, planning the use of land, air, and water, teaching of advanced engineering subjects, performing engineering surveys and studies, and the review and/or management of construction for the purpose of monitoring and/or ensuring compliance with drawings and specifications; any of which embraces such services or work, either public or private, in connection with any utilities, structures, buildings, machines, equipment, processes, work systems, projects, communication systems, transportation systems, and industrial or consumer products, or equipment of a control systems, communications, mechanical, electrical, hydraulic, pneumatic, chemical, environmental, or thermal nature, insofar as they involve safeguarding life, health, or property, and including such other professional services as may be necessary to the planning, progress, and completion of any engineering services.

Guidance:

Software engineers typically become specialists in various aspects of the practice of software engineering, based on their inclinations and experiences. There are four areas in which software engineers typically become specialists: 1) within particular domains; 2) within software engineering knowledge areas; 3) within product quality; and 4) within disciplines that support development and modification of software systems and products.

1. Specialties include expertise within particular domains, such as medical devices, automotive systems, information technology, scientific computing, or satellite navigation.
2. Within those domains, a practicing software engineer may become a specialist in one of the nine software engineering knowledge areas; for example in software construction or testing.
3. Alternatively, a software engineer may become a specialist in designing software to enhance product quality attributes such as safety, security, reliability, or ease of modification.
4. Supporting disciplines in which a software engineer might specialize include:
    - Configuration management
    - Quality assurance
    - Verification
    - Validation
    - Product sustainment
    - Process assessment and improvement

These specialty areas are combined in various ways. For example, a software engineer might specialize in process improvement for improving the processes used to verify and validate the safety of software within the automotive domain.

The scope of practice of an experienced software engineer often includes specialization beyond application of the knowledge and skills expected of all experienced practitioners; however, expertise in a specialty area should supplement, and not replace the knowledge and skills expected of all experienced software engineers.

Software project managers seeking licensure are expected to have requisite experience in one or more of the nine software engineering areas determined by the PAKS study, for example, in software design, construction or testing.

Section 5 - Scope of Software Engineering Practice Including Dependencies on System Engineering Activities

Relevant Section: *Model Law* 110.20, Definitions; A. Engineer; 5. Practice of Engineering; paragraph 1;

> **Practice of Engineering**—The term "Practice of Engineering," as used in this Act, shall mean any service or creative work, the adequate performance of which requires engineering education, training, and experience in the application of special knowledge of the mathematical, physical, and engineering sciences to such services or creative work as consultation, investigation, expert technical testimony, evaluation, planning, design and design coordination of engineering works and systems, planning the use of land, air, and water, teaching of advanced engineering subjects, performing engineering surveys and studies, and the review and/or management of construction for the purpose of monitoring and/or ensuring compliance with drawings and specifications; any of which embraces such services or work, either public or private, in connection with any utilities, structures, buildings, machines, equipment, processes, work systems, projects, communication systems, transportation systems, and industrial or consumer products, or equipment of a control systems, communications, mechanical, electrical, hydraulic, pneumatic, chemical, environmental, or thermal nature, insofar as they involve safeguarding life, health, or property, and including such other professional services as may be necessary to the planning, progress, and completion of any engineering services.

Guidance:

The practice of software engineering always occurs within a larger context because functioning software is always part of a larger system—even if only the processor upon which it is executed. Software may be implemented on a small digital device, such as a special purpose chip that is concurrently being developed as part of a larger system, or on a personal computer (PC) to implement a Web-based client-server application. Alternatively, a practicing software engineer may be a member of a team that develops software for a concurrent network of satellites being designed for communications.

There is, thus, a close connection between software engineering and other engineering activities. All engineers engage in synthesis and analysis, and attempt to develop and modify systems in a timely and economical manner. The primary distinction between software engineering and the traditional engineering disciplines arises from the ways in which software engineers apply their skills in the software engineering knowledge areas listed in Section 3 of this document. The distinction derives from the unique nature of software, which has no physical properties, as compared to the systems and products developed within the traditional engineering disciplines, but which may control physical devices.

Software engineers are sometimes included as members of systems-level teams, and engage in trade studies to determine optimal allocations of requirements, interfaces, and the resulting constraints on the various system components, including software. At other times, software engineers are recipients of system-level design decisions that are imposed on them.

An experienced software engineer should understand the role software plays within total systems, the similarities and the differences between software engineering activities and engineering activities in other engineering disciplines, and how to effective interact with other engineers when participating in engineering teams.

Section 6 - Standards of Practice

Annotates: *Model Law* 110.20, Definitions; A. Engineer; 5. Practice of Engineering; paragraph 1;

> **Practice of Engineering**—The term "Practice of Engineering," as used in this Act, shall mean any service or creative work, the adequate performance of which requires engineering education, training, and experience in the application of special knowledge of the mathematical, physical, and engineering sciences to such services or creative work as consultation, investigation, expert technical testimony, evaluation, planning, design and design coordination of engineering works and systems, planning the use of land, air, and water, teaching of advanced engineering subjects, performing engineering surveys and studies, and the review and/or management of construction for the purpose of monitoring and/or ensuring compliance with drawings and specifications; any of which embraces such services or work, either public or private, in connection with any utilities, structures, buildings, machines, equipment, processes, work systems, projects, communication systems, transportation systems, and industrial or consumer products, or equipment of a control systems, communications, mechanical, electrical, hydraulic, pneumatic, chemical, environmental, or thermal nature, or products that involve significant personal financial transactions or access to personal records, insofar as they involve safeguarding life, health, privacy, or property, and including such other professional services as may be necessary to the planning, progress, and completion of any engineering services.

Guidance:

There is a corpus of existing standards for software engineering. The two largest collections are maintained by ISO/IEC JTC 1/SC 7, Software and systems engineering, and the IEEE Software and Systems Engineering Standards Committee (S2ESC). Although the standards are labeled as "software engineering," they do not, in general, distinguish between the engineering of software and non-engineering development of software. Most of the described practices are equally applicable to both engineering and non-engineering development although engineering would imply a more rigorous application of the practices.

One standard is helpful in understanding the distinction. ISO/IEC 15026:1998 (currently being revised as ISO/IEC/IEEE 15026-3) describes a concept of integrity levels that is useful in delineating engineering versus non-engineering development. The standard may be applied to perform risk analysis to determine the elements of the system which have the greatest risk of consequence to life, health or property. Systems or system elements with the greatest risk would require the services of engineers. Lower consequence systems or elements would not. The standard defines the concept of integrity level but does not prescribe a set of levels; this is typically provided by industry-specific standards in a manner that is probably not unique to software systems.

ISO/IEC 15026 is intended to be applied in the context of a comprehensive set of processes describing the life cycle of a software system. ISO/IEC/IEEE 24748-1 describes exemplar life cycle models for software systems and explains that a life cycle consists of stages separated by decision gates. The activities of each stage are described as a collection of processes that serve to motivate, organize and prescribe the activities.

ISO/IEC/IEEE 12207 provides a suitable set of software life cycle processes.

ISO/IEC/IEEE 15288 provides a similar, but more abstracted, set of processes suitable for general systems. The life cycle processes can be used to define roles for organizations and individuals participating in software acquisition, supply, development, maintenance, operation (use), and disposal.

The various processes can be expected to produce information in addition to contributing to the software itself. Selected information must be managed as engineering documents. ISO/IEC/IEEE 15289 catalogues the various information items that are produced by the processes of 12207 and 15288.

None of the processes described by 12207 and 15288 and none of the information items described by 15289 can be regarded *a priori* as engineering versus non-engineering items. For any specific project, the integrity level required of the various elements may demand engineering attention. In addition, various processes, depending upon their contribution to the integrity level, may be regarding as engineering activities.

The determination of which processes are engineering activities and which documents are engineering artifacts should be performed during project planning. For this reason, project planning itself is an important engineering activity in any project which requires engineering services.

In addition to the standards described above, there are many additional standards providing additional detail or treating additional topics. Their selection for any given project must depend on the characteristics of the project. This, also, is a topic for project planning and may require engineering services.

Section 7 - Key Industry Sectors

Annotates: *Model Law* 110.20, Definitions; A. Engineer; 5. Practice of Engineering; paragraph 1;

> **Practice of Engineering**—The term "Practice of Engineering," as used in this Act, shall mean any service or creative work, the adequate performance of which requires engineering education, training, and experience in the application of special knowledge of the mathematical, physical, and engineering sciences to such services or creative work as consultation, investigation, expert technical testimony, evaluation, planning, design and design coordination of engineering works and systems, planning the use of land, air, and water, teaching of advanced engineering subjects, performing engineering surveys and studies, and the review and/or management of construction for the purpose of monitoring and/or ensuring compliance with drawings and specifications; any of which embraces such services or work, either public or private, ***in connection with any utilities, structures, buildings, machines, equipment, processes, work systems, projects, communication systems, transportation systems, and industrial or consumer products, or equipment of a control systems, communications, mechanical, electrical, hydraulic, pneumatic, chemical, environmental, or thermal nature,*** insofar as they involve safeguarding life, health, or property, and including such other professional services as may be necessary to the planning, progress, and completion of any engineering services.

Guidance:

It is important to understand that software can have no tangible physical consequences, including consequence to health, safety or welfare, if it is not part of a system which interacts with the physical world. The fact that software is a necessary part of a system implies that software always falls into some domain of application. However, software that does not control external devices can also affect the health, safety and welfare of the public. For example, software that controls access to a person's financial records (such as bank or retirement accounts) or personal information (such as health and identifying records) information can harm the public in the event of failure, whether by omission of functionality, error of functionality, or vulnerability to malicious attack. The consequences of software failure, hence its impact on health, safety or welfare, must be judged with respect to its domain of application. For example, software with identical functionality can be found in the fly-by-wire control systems of modern airplanes and in "flight simulator" games. The former is an application with high consequences, requiring engineering services; the latter is not.

The fact that software falls within some domain of application means that software engineering has a strong relationship with traditional engineering. The consequence of software failure can be judged via traditional engineering analysis of the consequences of failure of the system containing the software.

The engineering literature for many application domains is organized by industry sector. (In fact, most standardization is performed by industry-specific committees.) A list of industries could be as long as the list of ISO and IEC standards committees (a few hundred), but some may be noted as making particular use of software engineering. Examples include: defense, communication, electrical products, manufacturing, chemical, environmental, energy, security, medical, financial, and transportation. In addition, it should be noted that software utilized in the course of performing engineering calculation is also a domain of interest.

Section 8 - Relationship to Systems Engineering and Integrity Levels

Relevant Sections: *Model Law* 110.20 DEFINITIONS

> **5. Practice of Engineering—**The term "Practice of Engineering," as used in this Act, shall mean any service or creative work, the adequate performance of which requires engineering education, training, and experience in the application of special knowledge of the mathematical, physical, and engineering sciences to such services or creative work as consultation, investigation, expert technical testimony, evaluation, planning, design and design coordination of engineering works and systems, planning the use of land, air, and water, teaching of advanced engineering subjects, performing engineering surveys and studies, and the review and/or management of construction for the purpose of monitoring and/or ensuring compliance with drawings and specifications; any of which embraces such services or work, either public or private, in connection with any utilities, structures, buildings, machines, equipment, processes, work systems, projects, communication systems, transportation systems, and industrial or consumer products, or equipment of a control systems, communications, mechanical, electrical, hydraulic, pneumatic, chemical, environmental, or thermal nature, **insofar as they involve safeguarding life, health, or property**, and including such other professional services as may be necessary to the planning, progress, and completion of any engineering services.
>
> Design coordination includes the review and coordination of those technical submissions prepared by others, including as appropriate and without limitation, consulting engineers, architects, landscape architects, surveyors, and other professionals working under the direction of the engineer.
>
> Engineering surveys include all survey activities required to support the sound conception, planning, design, construction, maintenance, and operation of engineered projects, but exclude the surveying of real property for the establishment of land boundaries, rights-of-way, easements, and the dependent or independent surveys or resurveys of the public land survey system.
>
> A person shall be construed to practice or offer to practice engineering, within the meaning and intent of this Act, who practices any discipline or branch of the profession of engineering; or who, by verbal claim, sign, advertisement, letterhead, card, or in any other way represents the person to be a professional engineer, or through the use of some other title implies that the individual is a professional engineer or that the person is licensed or authorized under this Act; or who holds the person out as able to perform, or who does perform any engineering service or work or any other service designated by the practitioner which is recognized as engineering.

Guidance:

In cases where the software failure can lead to a significant risk to life, health or property then the development of the software falls under the practice of engineering and must be the responsibility of a licensed engineer with knowledge of software.

Software is usually a component of any complex system and hence the risks associated with software failure are a function of the risks of system failure and the role the software plays within the system design. Changes to the design of the system can change the level of risk associated with a software failure. The level of risk from the software will drive the requirements for engineering the software. It is because of these relationships that systems engineering and software engineering are intertwined.

Standard IEC/ISO 15026 defines the concept of system and software integrity levels. Integrity levels correspond to the level of risk associated with the system (or software component). The standard defines the concept of integrity levels but does not prescribe a set of levels; this is typically provided by industry-specific standards. The specific requirements on the software engineering process that result from each software integrity level are also typically provided by industry-specific standards.

In summary, it is the system engineering process that determines the role of a software component in a system and hence its integrity level. When the integrity level determination process establishes that the risk associated with the software is greater than some industry-established threshold, then the development of the software is considered to fall under the "practice of engineering" since it can impact life, health or property.

Section 9 – Responsibility and Fitness for Purpose/Use

Relevant Sections: *Model Law* 110.20 DEFINITIONS

> **5. Practice of Engineering—** The term "Practice of Engineering," as used in this Act, shall mean any service or creative work, the adequate performance of which requires engineering education, training, and experience in the application of special knowledge of the mathematical, physical, and engineering sciences to such services or creative work as consultation, investigation, expert technical testimony, evaluation, planning, design and design coordination of engineering works and systems, planning the use of land, air, and water, teaching of advanced engineering subjects, performing engineering surveys and studies, and the review and/or management of construction for the purpose of monitoring and/or ensuring compliance with drawings and specifications; any of which embraces such services or work, either public or private, in connection with any utilities, structures, buildings, machines, equipment, processes, work systems, projects, communication systems, transportation systems, and industrial or consumer products, or equipment of a control systems, communications, mechanical, electrical, hydraulic, pneumatic, chemical, environmental, or thermal nature, insofar as they involve safeguarding life, health, or property, and including such other professional services as may be necessary to the planning, progress, and completion of any engineering services.
>
> Design coordination includes the review and coordination of those technical submissions prepared by others, including as appropriate and without limitation, consulting engineers, architects, landscape architects, surveyors, and other professionals working under the direction of the engineer.

Guidance:

It is ultimately the decision of each jurisdiction to determine what forms of software engineering practice should be licensed. Because software is ubiquitous in many systems designed today, the question becomes one of how closely the software interacts with systematic aspects that could result in a threat to public health, safety, or welfare. Almost all software could in some way tangentially contribute to such risks. Therefore it is impossible to classify software systems as either "requires a professional engineer" or "does not require a professional engineer." Instead, it is suggested that each board resolve this issue by answering a set of questions that first appeared in an article authored by Phil Laplante and Mitch Thornton - "When do Software Systems Need to be Engineered," *Today's Engineer*, July 2011, available at http://www.todaysengineer.org/2011/Jul/licensure.asp:

- Q1. Does the software control a device or devices that could directly inflict harm to a human being if there was a malfunction?
- Q2. Does the software put the assets of an individual or corporate entity at risk beyond the normal amount of risk assumed in everyday business transactions?
- Q3. Does the software expose identifying information of an individual or a corporate entity that would violate any federal, state or local laws (e.g. HIPPA, FERPA)?
- Q4. Does the software interact with other systems in way that directly satisfies 1-3 above*?

* Note: Software engineers need to exercise constant vigilance on the operational security picture and never design, develop, or maintain software in a vacuum removed from the front lines. Development teams usually do their risk assessment based on the impact of **their** application being compromised. A professional should consider the OPERATIONAL environment in which the software will operate.

Section 10 - Approvals, Delegations, and Exemptions

Relevant Sections: *Model Law* 170.20 C. Exemption Clause

> This Act shall not be construed to prevent the following:
>
> A. Other Professions—The practice of any other legally recognized profession.
>
> B. Contingent License—A contingent license may be issued by the board or board administrator to an applicant for comity licensure if the applicant appears to meet the requirements for comity licensure. Such a contingent license will be in effect from its date of issuance until such time as the board takes final action on the application for comity licensure. If the board determines that the applicant does not meet the requirements for issuance of a comity license, the contingent license shall be immediately and automatically revoked upon notice to the applicant and no comity license will be issued.
>
> C. Employees and Subordinates—The work of an employee or a subordinate of an individual holding a certificate of licensure under this Act, or an employee of an individual practicing lawfully under Subsection B of this section, provided such work does not include final engineering or surveying designs or decisions and is done under the responsible charge of and verified by an individual holding a certificate of licensure under this Act or an individual practicing lawfully under Subsection B of this section.

Guidance:

Approvals and Delegation –

As with other types of engineering, work done by an employee of a licensed software engineer under direct supervision is considered the work of the engineer. Subsection C of 170.20 exempts such an employee from licensure requirements if their work does not include final engineering products. If a licensed software engineer has other engineers or professionals working under his or her direct supervision, those professionals may do engineering work. Subsection A also exempts the work of other legally recognized professions. In the software development universe, there are many different roles for software professionals, which are legally recognized. The definitions of roles and responsibilities related to software development and software engineering continue to evolve and get greater clarification. An engineer in responsible charge of an engineering project is a critical component even if other professionals are involved. All engineers, including software engineers are charged with and responsible for protecting the health, safety and welfare of the public.

Exemptions –

The NCEES Model law does not address other exemptions from licensure requirements, although most state laws have them. They cover many engineers working in certain organizations. One of the most applicable and widely used is commonly referred to as "Industrial Exemptions." Many states have industrial exemptions which generally do not require practicing engineers to be licensed in certain circumstances. Typically, the industrial exemptions were created to place the responsibility for protecting the health, safety and welfare of the public on a private corporation or other organization instead of an individual engineer. A private company that sells an "off the shelf" product is responsible for the effects it has on the health, safety and welfare of the consumer of the product. Industrial exemptions apply in most states and cover engineer employees in a wide range of manufacturing, oil and gas and chemical production industries. These industrial exemptions would also apply to certain software development environments. Engineers working for a company designing commercial, "off the shelf" software would likely not be required to be licensed in states with industrial exemptions. The same software engineer or engineering firm, who offered engineering services as a consultant or practiced independently may require licensure.

Some examples of industrial exemptions are:

Texas

1001.057. Employee of Private Corporation or Business Entity

> (a) This chapter shall not be construed to apply to the activities of a private corporation or other business entity, or the activities of the full-time employees or other personnel under the direct supervision and control of the business entity, on or in connection with:
>
> > (2) activities related only to the research, development, design, fabrication, production, assembly, integration, or service of products manufactured by the entity.

(d) For purposes of this section, "products manufactured by the entity" also includes computer software, firmware, hardware, semiconductor devices, and the production, exploration, and transportation of oil and gas and related products.

California

6747. Exemption for industries

(a) This chapter, except for those provisions that apply to civil engineers and civil engineering, shall not apply to the performance of engineering work by a manufacturing, mining, public utility, research and development, or other industrial corporation, or by employees of that corporation, provided that work is in connection with, or incidental to, the products, systems, or services of that corporation or its affiliates.

(b) For purposes of this section, "employees" also includes consultants, temporary employees, contract employees, and those persons hired pursuant to third-party contracts.

New York

1. 7208. Exempt persons.

This article shall not be construed to affect or prevent the following, provided that no title, sign, card or device shall be used in such manner as to tend to convey the impression that the person rendering such service is a professional engineer or a land surveyor licensed in this state or is practicing engineering or land surveying:

k. The practice of engineering by a manufacturing corporation or by employees of such corporation, or use of the title "engineer" by such employees, in connection with or incidental to goods produced by, or sold by, or non-engineering services rendered by, such corporation or its manufacturing affiliates;

Florida

471.003 Qualifications for practice; exemptions.

(2) The following persons are not required to be licensed under the provisions of this chapter as a licensed engineer:

(c) Regular full-time employees of a corporation not engaged in the practice of engineering as such, whose practice of engineering for such corporation is limited to the design or fabrication of manufactured products and servicing of such products.

Section 11 - Competence

Annotates: *Model Law* 130.10, General Requirements for Licensure; C. Professional Engineer; 1. As a Professional Engineer; b. Licensure by Examination

> The following individuals shall be admitted to an ***8-hour written examination in the principles and practice of engineering*** and, upon passing such examination and providing proof of graduation, shall be licensed as a professional engineer, if otherwise qualified:
>
> (1) An engineer intern with a bachelor's degree in engineering and with a specific record of ***4 years or more of progressive experience on engineering projects of a grade and a character which indicate to the board that the applicant may be competent to practice engineering***

Guidance:

In one sense, the requirements for licensing a software engineer are no different than the requirements for any other engineering specialty—an EAC-ABET accredited degree, two 8-hour examinations and 4 years of qualifying experience. NCEES is currently developing a suitable examination for software engineering. Judging qualifying experience may present some practical difficulties. Because similar software development activities may or may not be judged as engineering activities based on the consequences of the system to which they contribute, a simple listing of work roles and work experience does not suffice to indicate engineering experience. In judging qualifying experience, one must consider the application domain of the system, the potential consequences of the system on health, safety and welfare, the contribution of the software element to the function of the system, and the contribution of the experienced activities to the development of the software element. Well-engineered systems should produce documentation describing these relationships and such documentation could be helpful in evaluating experience.

The organizational relationship of the candidate to licensed engineers on the project may be indicative of the nature of the experience and the documented results of activities performed by the candidate may also be helpful. The level of engineering responsibility exercised by the candidate may also be indicative.

Section 12 – Deliverables or Work Products

Relevant Section: *Model Law* 140.10 Certificates of Licensure, Seals

A. The board shall issue to any applicant for licensure as a professional engineer or professional surveyor who, in the opinion of the board, has met the requirements of this Act, a certificate of licensure giving the licensee proper authority to practice his or her profession in this jurisdiction. The certificate of licensure for a professional engineer shall carry the designation "Professional Engineer" and for a professional surveyor, "Professional Surveyor." It shall give the full name of the licensee with licensure number and shall be signed by the chairperson and the board administrator under the seal of the board.

B. The certificate of licensure shall be prima facie evidence that the individual named thereon is entitled to all rights and privileges and is bound by all responsibilities of a professional engineer or a professional surveyor while the said certificate of licensure remains active and unrestricted.

C. Each licensee hereunder must, upon licensure, obtain a seal as described in Section 110.20 K of this Act. Documents must be sealed, signed, and dated in accordance with the Rules.

D. The board shall issue to any applicant for certification as an engineer intern or surveyor intern who, in the opinion of the board, has met the requirements of this Act, an enrollment document as engineer intern or surveyor intern, which indicates that his or her name has been recorded as such in the board office. The engineer intern or surveyor intern enrollment document does not authorize the holder to practice as a professional engineer or a professional surveyor.

Guidance:

There are numerous artifacts that may be produced before, during and after the software code is written, tested, integrated into the operational platform, and delivered. These artifacts include, but are not limited to:

1. System and/or software requirements specifications
2. Software architectural specifications
3. Software design specifications
4. Software source code
5. Software test plans
6. Software test acceptance documentation
7. Software integration plans
8. User manuals
9. Configuration audit information
10. Software maintenance plans
11. Coding standards
12. Change request forms
13. Defect and trouble reports

This is by no means an exhaustive list. Each of these artifacts and related types may contain information that is essential to the safe specification, apportionment, design, construction, testing, change control, and maintenance of software and therefore could be considered as "design" artifacts subject to sealing by a professional engineer. Not all artifacts apply in all cases, however. This list should be considered with the definitions and standards of practice in Section 6 of this document. As stated in Section 6, the determination of which documents are engineering artifacts should be performed during the planning of the particular project.

"Sealing" of engineering documents related to software is a particularly difficult concept compared to other types of engineering. The documents and artifacts listed above may have no unique physical existence upon which to affix an engineer's seal. Each state mandates its own sealing requirements, but most require the image of a PE's unique seal, image of the PE's unique signature and the date. The purposes of the seal for traditional engineering documents are to freeze the document for auditability and authenticate it as the work of a qualified Professional Engineer. Ultimately, "sealing" a document for software engineering would serve the same purposes.

For software and associated artifacts, a specific version would be "sealed". This could be accomplished with a physical cover letter that specifies version and includes the typical signed and dated seal. For states that allow for electronic seals and signatures, this could be accomplished with an electronic version of a coversheet that specifies version and includes the appropriate information regarding the identification of the engineer. In addition to the images of seal and signatures, encryption techniques should be used to secure the document and provide auditability. Some software development will include Configuration Management documentation, which could also perform the same functions.

2001 L Street, NW Suite 700
Washington, D.C. 20036
+1 202 785 0017
www.ieeeusa.org