

## GUIDELINE

# Professional Engineers Using Software-Based Engineering Tools

### CONTRIBUTORS

Eric Brown, P. Eng.  
Colin Cantlie, P. Eng.  
Norm Fisher, P. Eng.  
Jeremy Jackson, P. Eng.  
Tibor Palinko, P. Eng.  
Daniel Tung, P. Eng. (Chair)

**Notice:** The Professional Standards Committee has a policy of reviewing guidelines every five years to determine if the guideline is still viable and adequate. However, practice bulletins may be issued from time to time to clarify statements made herein or to add information useful to those professional engineers engaged in this area of practice. Users of this guideline who have questions, comments or suggestions for future amendments and revisions are invited to submit these to PEO using the form provided in Appendix 2.

May 2009 draft

## **Table of Contents**

1. PEO MANDATE AND CRITERIA FOR GUIDELINES
  2. PREFACE
  3. PURPOSE AND SCOPE OF GUIDELINE
  4. INTRODUCTION
  5. USING SOFTWARE-BASED ENGINEERING TOOLS
    - 5.1 Choosing the right software
    - 5.2 Understanding the software in an engineering context
    - 5.3 Training and support for the software user
    - 5.4 User's responsibility for quality assurance
    - 5.5 Configuration management for software
    - 5.6 Traceability for data and results
    - 5.7 Quality control
  6. VERIFICATION
  7. ISSUES RELATED TO SOFTWARE DERIVED INFORMATION
- APPENDIX 1: DEFINITIONS
- APPENDIX 2: AMENDMENT AND REVISION SUBMISSION FORM
- APPENDIX 3: PEO PROFESSIONAL PRACTICE GUIDELINES

## 1. PEO MANDATE AND CRITERIA FOR GUIDELINES

Professional Engineers Ontario produces guidelines for the purpose of educating both licensees and the public about standards of practice. This is done to fulfill PEO's legislated objectives. Section 2(4)2 of the *Professional Engineers Act* states: "For the purpose of carrying out its principal object", PEO shall "establish, maintain and develop standards of qualification and standards of practice for the practice of professional engineering". The Association's Professional Standards Committee is responsible for developing practice standards and preparing guidelines.

Professional Engineers Ontario produces guidelines to meet the following objectives, which were used to develop the content of this document.

1. Guidelines are intended to aid engineers in performing their engineering role in accordance with the *Professional Engineers Act* and Regulations 941/90 and 260/08.
2. Guidelines are intended to define processes required by regulatory, administrative or ethical considerations associated with specific professional services provided by engineers. They do not aim to be short courses in an engineering subject.
3. Guidelines provide criteria for expected practice by describing the required outcome of the process, identifying the engineer's duty to the public in the particular area of practice, and defining the relationships and interactions between the various stakeholders (i.e. government, architects, other engineers, clients).
4. Guidelines add value to the professional engineer licence for practitioners and for the public by establishing criteria for standards of best practice.
5. Guidelines help the public to understand what it can expect of engineers in relation to a particular task within the practice of professional engineering. By demonstrating that the task requires specialized knowledge, higher standards of care, and responsibility for life and property, guidelines help reinforce the public perception of engineers as professionals.

This guideline is not intended to establish a "one method of practice for all" approach to the practice of professional engineering. This guideline is not intended to replace a practitioner's professional judgment when providing professional engineering services. Subject to provisions in the guideline that incorporate professional conduct requirements or legal requirements, a decision by a practitioner not to follow the guideline will not, in and of itself, indicate that a member has failed to maintain an acceptable standard of work. On the other hand, following the guideline may not ensure that a member has provided services conforming to an acceptable standard. Determining whether a practitioner's service is acceptable will depend upon the circumstances of each case.

See Appendix 3 for a list of PEO professional practice guidelines.

## **2. PREFACE**

Professional Standards Committee formed a subcommittee of practitioners from a variety of practice areas who had experience providing professional engineering services with the assistance of software. This group was asked to address questions regarding the proper role for and responsibilities incurred by professional engineers using commercial, free source and practitioner developed software. The subcommittee was also instructed to prepare a guideline providing recommendations to all professional engineers depending on software as an aid in providing engineering services.

The subcommittee met for the first time on May 26, 2006, and submitted a completed draft of this document to the Professional Standards Committee for approval on May 14, 2009.

Following practitioner consultations, the final draft was approved by Council at its meeting on XXX XX, 2009.

## **3. PURPOSE AND SCOPE OF GUIDELINE**

Most professional engineers rely on software to undertake tasks such as calculations, modeling, and optimization analysis as part of their engineering services or to provide information used as the basis for their decisions, judgments and opinions.

This guideline suggests standards for the use of computer programs in engineering work that are reasonably necessary for the protection of the public. The practices, procedures and controls presented here must be tailored to each project's specific requirements. They are strongly recommended, but are not considered mandatory by the association.

This guideline does not deal with development of software. Situations involving software as the output of an engineering design process is the subject of a separate guideline.

Note: References in this guideline to professional engineers apply equally to temporary licence holders, provisional licence holders and limited licence holders.

## **4. INTRODUCTION**

The practice of professional engineering has become increasingly reliant on computers, and engineers use many computer programs that incorporate engineering principles as aids in carrying out their assignments. This software generally provides numerical or graphical output that the engineer will use as the basis of decisions. In many cases software provides complete design information such as pipe or wire sizing, truss layouts, or equipment selection that the engineers can utilize directly in their work. Parametric design software such as that available for ductwork and piping design can produce complete drawings with no operator interaction beyond inputting initial conditions.

Every practitioner has responsibility for all aspects of the design or analysis they incorporate into their work whether it is done by an engineer-in-training, a technologist or a computer program. Therefore, practitioners are advised to always use the data obtained from engineering software judiciously and use it only after submitting results to

vigorous checking process. This guideline explains the steps needed to fulfill the practitioner's due diligence obligations.

Due diligence is the effort expected to be made by an ordinarily prudent or reasonable party to avoid harm to another party. A practitioner's due diligence is best demonstrated by taking an organized approach to ensuring all potential sources of error and omission are assessed and, if necessary, corrected before any action is taken. The procedures and processes described in this guideline are an example of best practices for due diligence in situations where practitioners rely on software tools.

All practitioners must have an acceptable knowledge of the engineering principles involved in all the work they undertake. Article 72(2)(h), O. Reg. 941 identifies one criteria of professional misconduct as "undertaking work that practitioner is not competent to perform by virtue of the practitioner's training and experience". Using software to automate part of the work does not relieve practitioners of the obligation to provide services only in their area of competency.

## **5. USING SOFTWARE-BASED ENGINEERING TOOLS**

This guideline deals with situations where software is used to provide assistance to professional engineers in performing the engineering activities included in the definition of the practice of professional engineering given in Section 1 of the *Professional Engineers Act*. The listed activities are designing, composing, evaluating, advising, reporting, directing or supervising. This guideline refers to the use of engineering software as a contributor to any of these activities.

### **5.1 Choosing the right software**

As part of their due diligence obligations practitioners have responsibility to accept or reject software based on their own assessment of the compatibility and viability of the software to the task at hand. Engineering practitioners must understand the limitations of the software tools proposed to be used for an engineering project and carefully consider whether the software is appropriate for the purpose intended. A practitioner should decide to use a software-based engineering tool only if he or she has a high degree of confidence that it is the appropriate technical means for accomplishing the task.

Even the best software will provide incorrect results if the engineer is either using it incorrectly or is using software that is inappropriate for the engineer's needs. Every practitioner should consider, prior to use, whether the software to be used is the right product for the purpose and that it can perform this work reliably. If the software used is not appropriate the practitioner may be relying on faulty data. For example, software provided by an equipment supplier might rely on technical information specific to that equipment and for that reason the outputted data may not be generally applicable. Using this data may result in an ill-conceived and problematic design.

Before purchasing or using software practitioners should assure themselves that it will do what they need. By reading reviews and talking to other users, practitioners should be able to determine the software's capabilities and whether it is dependable. However, practitioners (or someone in the organizations employing practitioners) should be able to justify the selection of each software tool.

The first step in assessing whether a certain software tool is right for the task is to document requirements for the software tool and then validate that the chosen tool actually meets these requirements. Some of the questions to ask yourself include: Can all input data be seen on the page? Or do you see data only as it is entered? Does the software provide data entry checking to ensure that the data entered is within reasonable bounds? Does the software provide a copy of the input data in the output display or printout? Engineers should avoid software that does not provide a means for verifying and recording input data.

Does the software have defaults for data not supplied by the user? Are the defaults reasonable? Does the software allow the practitioner to modify all relevant data or does it appear that certain constants are fixed by the developer. In other words, does the software offer the practitioner sufficient control or is the practitioner held hostage to decisions made by the developer?

## 5.2 Understanding the software in an engineering context

Knowing how to use the software properly is crucial for getting correct answers. Practitioners should become familiar with all aspects of the software prior to using it for design or analysis purposes.

Engineering programs are based upon or include assumptions, limitations, interpretations and judgments on engineering matters that were made by or on behalf of the user when the program was first developed. It is often difficult to determine, just by using a program or by being given a description of its function how the software deals with the engineering principles and technical information it incorporates. Engineers should become familiar with the engineering principles, equations, models, algorithms and assumptions used in the software. Some developers provide manuals or white papers containing detailed explanations of the software's underlying structure. Practitioners should acquire and read these documents.

## 5.3 Training and support for the software user

Practitioners should also consider taking courses available through the supplier or a third party trainer. When the developer does not supply a user manual or the manual is not comprehensive the engineer should obtain third party texts, if these are available. Make use of informal sources of expert advice. For example, users of many common software applications have established support networks such as forums and user groups that share experiences and information regarding the software. These groups can provide warnings about software bugs, help resolving application problems, and suggestions for improving the software and its use.

## 5.4 User's responsibility for quality assurance

Quality assurance is a process meant to ensure that (a) software is installed and configured correctly; and (b) that all users are using the software correctly. The process will involve checking configuration management records and testing based on standardized input.

The quality assurance procedure should be performed every time software or relevant hardware is updated or reconfigured. This is done by qualifying each new version of the

software against the standard set of data and observing any deviations from output generated by previous versions. In order to substantiate results practitioners should keep records of input data, preferably a record supplied by the software either as an electronic data file or as a printout and check output from newer software version with that obtained from earlier, verified versions.

When verifying updated versions of previously used software, testing may concentrate on those features changed or added in the new version. However, practitioners should not assume that previously existing features continue to function correctly.

The practitioner should test all problematic situations that arose in the past. This will require the keeping of logs reporting software performance and user observations of problems experienced and limitations.

In larger organizations, software choice, testing and verification may be undertaken by other qualified people rather than by every practitioner. In these cases a practitioners need only assure themselves that such corporate practices exist and are responsibly executed and documented. The practitioner needs only to review that record rather than be individually responsible for each of the QA tasks described in this section.

#### 5.5 Configuration management for software

Configuration management is record-keeping related to the maintenance, upgrading, alteration and problems experienced with the hardware and software used by the organization. Configuration management consists of four tasks:

- Identification – this is the selecting, specification, identification of all components and their inter-relationships and making provision to record and retain this information.
- Control – this is the management of each identified configuration item, specifying who is authorized to ‘change’ it, and ensuring only authorized and identifiable configuration items are accepted.
- Status – this task is the recording of the status of all configuration Items in the database, and the maintenance of this information.
- Verification – this task involves reviews and audits to ensure the recorded information is accurate.

Keeping these records will ensure that the practitioner knows the version currently being used and whether everyone in the organization or project team is using the same version. The records will confirm whether this version been verified for use.

#### 5.6 Traceability for data and results

Even if the software has some form of input checking, it cannot determine if the information provided is the correct information for the project. The practitioners should know which version of input data was used to generate every set of output. It is also important to know which version of the software produced the output data.

## 5.7 Quality control

Quality control procedures should be implemented each time software is used to carry out engineering activities. It is the responsibility of engineers to ensure that their designs or the opinions they provide in reports are correct. This includes designs and opinions based on software derived data. Output data should be checked thoroughly enough that the engineer is reasonably assured that the data is correct. This may involve manual calculation of a random selection of the output data or comparison with data obtained for past projects of a similar nature. Defective output discovered while using software-based tools should be examined closely to determine the implications on suitability for use of the tools. Practitioners who are not directly responsible for a firm's QA/QC or management/configuration processes should, after discovering 'defective' output where the nature of the defect is not obviously user error, notify the responsible persons.

## 6. VERIFICATION

Professional engineers are responsible for the accuracy, completeness and acceptability of all aspects of services they provide including information obtained using software. Given the increasing reliance on computer software, the engineer should ensure that a comprehensive verification of the software's performance exists.

Widely used, industry standard software undergoes ongoing verification by the regular users who routinely apply results obtained from the software to their engineering work. Each completed project based on the software is a testament to its applicability. In all cases, the engineer should establish and conduct suitable tests to determine whether the software performs as it is required to do.

Software verification is not the same as checking the output of a run to see if is reasonable. Checking of output after each run is necessary in order to ensure that the data was entered correctly for the particular engineering project underway. The checking process is a quality assurance measure employed to ensure that the output of a specific run is reliable and can be used for design purposes. This process assumes that the software operates correctly and, therefore, problematic output is caused by incorrect use, faulty input data, or a design that falls outside the bounds of the software's reliability.

Verification, on the other hand, is an evaluation, conducted outside the process of design and analysis for specific projects, in order to guarantee, prior to use, that the software always provides correct output. Software is verified by comparing the output, for well defined input conditions, against data known to be correct based on either actual results from real-life situations or thoroughly checked manual calculations.

The verification process can also be used to ensure that every person using the software does so correctly or to compare various software products to determine which is the most reliable.

## 7. ISSUES RELATED TO SOFTWARE DERIVED INFORMATION

Output from a computer program, like all working notes and calculations are the property of the engineer and generally do not need to be provided to the client or submitted as part of regulatory approval processes unless required by law or contract.

In those cases where output data must be supplied to persons outside the organization employing the practitioner the data should be distributed either as hard copies or electronic files bearing the seal and signature of the professional engineers who prepared or supervised the preparation of the input data and checked the output data. See the PEO guideline *Use of the Professional Engineer's Seal* for further information regarding the proper use of the seal.

## **APPENDIX 1 DEFINITIONS**

For the purposes of this guideline:

### **Configuration Item**

An object that is treated as a self contained unit for the purposes of identification and change control. All configuration items (CIs) should be uniquely identified by codes and version numbers. CIs may be a single module such as a monitor or tape drive, or more complex items, such as a complete system.

### **Configuration Management**

Process and procedures for maintaining a database containing details of the computer elements that are used in the provision of the organization's services. More than just an 'asset register', it will contain information that relates to the maintenance, movement, and problems experienced with the elements in the database.

### **Engineering Activities**

Activities included in the definition of the practice of professional engineering given in Section 1 of the *Professional Engineers Act*. The listed activities are designing, composing, evaluating, advising, reporting, directing or supervising.

### **Quality Assurance**

Refers to a managerial function involving all the planned and systematic activities implemented within the quality management system to provide confidence that the project will satisfy the relevant quality standards.

### **Quality Control**

An inspection function using quality control tools that involves monitoring specific project results to determine if they comply with relevant quality standards, and identifying ways to eliminate causes of unsatisfactory results. It is focused on the deliverables and involves procedures that determine if specified quality is attained.

### **Validation**

Refers to the process of determining the correctness of a deliverable with respect to the user's needs.

### **Verification**

Refers to the process of reviewing, inspecting, and testing of a software tool under standardized conditions to determine whether there is adequate assurance that the output of such tool will conform to the specified requirements.

**APPENDIX 2: AMENDMENT AND REVISION SUBMISSION FORM**

**Guideline:**

---

---

**Statement of proposed amendment or revision:**

---

---

---

---

---

**Reason:**

---

---

---

---

---

---

---

**Submitted by:**

---

**Date:**

---

**Mail:** Professional Engineers Ontario  
1000- 25 Sheppard Avenue West  
Toronto ON M2N 6S9  
Attention: Bernard Ennis, P.Eng., Manager, Standards and Practice

**Fax:** (416) 224-8168 or (800) 268-0496

**Email:** bennis@peo.on.ca

## **APPENDIX 2 PEO PROFESSIONAL PRACTICE GUIDELINES**

1. Acting as Contract Employees (2001)
2. Acting as Independent Contractors (2001)
3. Acting Under the Drainage Act (1988)
4. Acoustical Engineering Services in Land-Use Planning (1998)
5. Building Projects Using Manufacturer-Designed Systems & Components (1999)
6. Commissioning Work in Buildings (1992)
7. Communications Services (1993)
8. Engineering Services to Municipalities (1986)
9. Environmental Site Assessment, Remediation and Management (1996)
10. General Review of Construction as Required by the Ontario Building Code (2008)
11. Geotechnical Engineering Services (1993)
12. Guideline to Professional Practice (1998)
13. Human Rights in Professional Practice (2000)
14. Land Development/Redevelopment Engineering Services (1994)
15. Mechanical and Electrical Engineering Services in Buildings (1997)
16. Professional Engineer as an Expert Witness (1997)
17. Professional Engineer's Duty to Report (1991)
18. Project Management Services (1991)
19. Reports for Pre-Start Health and Safety Reviews (2001)
20. Reports on Mineral Properties (2002)
21. Roads, Bridges and Associated Facilities (1995)
22. Selection of Engineering Services (1998)
23. Solid Waste Management (1993)
24. Structural Engineering Services in Buildings (1995)
25. Temporary Works (1993)
26. Transportation and Traffic Engineering (1994)
27. Use of Agreements between Client and Engineer for Professional Engineering Services (including sample agreement) (2000)
28. Use of the Professional Engineer's Seal (2008)
29. Professional Engineers Using Software-Based Engineering Tools