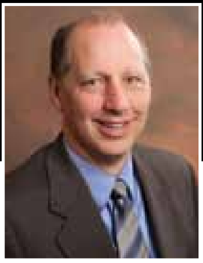
Editorial

The "Ins" and "Outs" of the Software Black Box



By Andrew Rauch, CASE Chair

Last spring, I had the opportunity to attend one of the several terrific presentations that were part of the CASE Risk Management Convocation at the 2013 Structures Congress. This session was presented by James Parker and Pedro Sifre of Simpson, Gumpertz, and Heger, and I thank them for allowing me to present a portion of it to a wider audience.

The use of design software is in integral part of the structural engineering design process. None of us can imagine what our profession would be like without it. At the same time, it presents challenges and concerns to those who are responsible for the operation of an engineering firm. The attendees at this session discussed some of those challenges, including staff skills and training, the black box aspects of software, documentation of software results, and the delegation of design and code interpretation to software companies.

Are we training our staff to use software appropriately? Are we giving up an opportunity for young engineers to develop skills needed to conceive and implement structural designs by allowing them to extensively use software for design? As a young engineer, I learned design through repetition, reading code requirements and applying them in preparing calculations. (How many of you eventually memorized the member properties of some of the common beam sizes?) I began to understand what the expected results should be prior to performing the calculations. Today's engineers need and use very different skills. They need to learn how to use software effectively, to learn how to properly build a structural model, and to learn how to make their design model interface with building information models. When and how do they develop a "feel" for the structure and intuition about a design that tells them if their design is reasonable?

The situation may arise where an engineer is using software to design a system they may not have previously designed. The software is able to provide design results for that structure, but has the engineer developed the skills to determine if the design results are correct or reasonable? Does the engineer have the skills necessary to approximate the design to verify the software results? Obviously, in this situation, the engineer needs a significant amount of oversight.

Structural engineering software can also be a black box. How often have you heard the explanation "that's what the output said" in response to a question about a design result? How does the program handle design conditions such as unbraced length, cracked member stiffness, or the algorithm for selecting the number of shear studs on a composite beam? Often, the manual provides little information to help the engineer determine what process the software is using. Are we deferring code interpretation and some of our quality assurance to the software provider?

The question we must ask ourselves is how much of our design skill and interpretation do we want to delegate to software companies?

Documentation of design is another issue. Have you ever been looking for design information to answer a question and found no written calculations? You try to find a result from the software, only to find several versions of the model with no clear indication of which one is the most current or what the different models signify. Young engineers will sometimes use the "brute force" method of design, using the computer to run multiple iterations. When it comes time to provide written documentation, suddenly there are pages and pages of calculation for a design problem that could have been designed much more simply. Are the requirements for computer analysis and design documentation procedures a part of your office policies and procedures?

The final question posed at this session asked how the profession should react. Should one (or all) of the structural engineering organizations provide reviews and vetting of software? Should we leave software verification to the purchasers and users, and let market forces drive software quality? Should the structural engineering organizations work with authorities having jurisdiction to demand certification or verification of software? While it would be nice to have third-party software verification, that is a Herculean task for structural engineering organizations that are run primarily by volunteers. For now, the consensus of those in attendance was to let market forces drive the quality.

The question we must ask ourselves is how much of our design skill and interpretation do we want to delegate to software companies? To our knowledge, there are no standards or requirements for software producers to check and verify their software. Writers of software codes are not required to be licensed to work under the direction of a licensed engineer. Our experience has been that every software program we have purchased or licensed has had some kind of error or bug that caused it not to work properly. How are we as individuals and as a profession going to react?▪

Andrew Rauch is a principal with BKBM Engineers in Minneapolis, MN and is responsible for overseeing their quality assurance and risk management programs. He is the current chair of the CASE Executive Committee. He can be reached at ***arauch@bkbm.com****.*