new york state society of

NYSSCPA

certified public accountants

Home of the Trusted Professional
3 park avenue, at 34th street, new york, ny 10016-5991
212.719.8300 • fax 212.719.3364
www.nysscpa.org

August 7, 2008

Ms. Sharon Macey
American Institute of Certified Public Accountants
1211 Avenue of the Americas
New York, New York 10036-8775

By e-mail: smacey@aicpa.org

**Re: Proposed Statement on Standards for Attestation Engagements; An
Examination of an Entity's Internal Control Over Financial Reporting That Is
Integrated With an Audit of Its Financial Statements (Supersedes AT Section 501,
Reporting on an Entity's Internal Control Over Financial Reporting)**

Dear Ms. Macey:

The New York State Society of Certified Public Accountants, representing 30,000
CPAs in public practice, industry, government and education, submits the following
comments to you regarding the above captioned exposure draft. The NYSSCPA thanks
the AICPA for the opportunity to comment.

The NYSSCPA's Technology Assurance Committee deliberated the exposure
draft, in particular sections 154 – 159, and drafted the attached comments. If you would
like additional discussion with us, please contact Bruce I. Sussman, Chair of the
Technology Assurance Committee, at (973) 422-7151, or Ernest J. Markezin, NYSSCPA
staff, at (212) 719-8303.

Sincerely,

Sharon Sabba Fierstein

Sharon Sabba Fierstein
President

Attachment

COMMENTS ON SECTIONS 154 – 159 OF PROPOSED STATEMENT ON STANDARDS FOR ATTESTATION ENGAGEMENTS; *AN EXAMINATION OF AN ENTITY'S INTERNAL CONTROL OVER FINANCIAL REPORTING THAT IS INTEGRATED WITH AN AUDIT OF ITS FINANCIAL STATEMENTS* **(SUPERSEDES AT SECTION 501,** *REPORTING ON AN ENTITY'S INTERNAL CONTROL OVER FINANCIAL REPORTING)*

August 7, 2008

Principal Drafter

Yigal Rechtman

New York State Society of Certified Public Accountants

Comments on Sections 154 – 159 of AICPA Exposure Draft of a
Proposed Statement on Standards for Attestation Engagements
An Examination of an Entity's Internal Control Over Financial Reporting That Is
Integrated with an Audit of Its Financial Statements
(Supersedes AT Section 501, *Reporting on an Entity's Internal Controls Over Financial Reporting*)

The Society's Technology Assurance Committee deliberated sections 154 – 159 of the Exposure Draft (ED) and prepared the following comments. We wish to thank the AICPA for the opportunity to comment.

Background

The automation of internal controls has become a substantial portion of the operations of companies, both large and small. Systems that we formerly referred to as "electronic data processing" (EDP) were serial in nature and simplistic in operation. Over decades, EDP has evolved into a multi-platform, complex environment. Modern information technology (IT) environments require auditors first to obtain a thorough understanding and second to perform detailed analyses in order to be assured that such systems are well suited for the function they are designed to serve, and are operating as designed.

Legacy EDP systems now represent only basic sources of information for financial reporting and management decision-making. Current IT systems are complex, multi-dimensional environments that can no longer be evaluated using a simple input-process-output paradigm. Substantial consideration and analysis is required of the attestation procedures that assess the risks and operational effectiveness of IT environments.

Comments on Sections 154 – 159

We have the following comments on sections 154 – 159 of the ED:

Section 154 discusses an automated application that is not subject to breakdowns due to human failure. However, as discussed below, such systems rarely exist because of the complexity of software that is critical to controls over financial reporting.

Section 154 speaks to a benchmarking strategy which we consider to be the weakest strategy in the spectrum of alternatives for evaluation of software systems. The reason the benchmarking strategy is ineffective in many situations is that most software applications that are critical for internal control over financial reporting are complex.

Complex software is multi-modular software that has many factors that affect the method by which the software receives, processes, stores and produces the information. Complex software involves a large number of variables, each of which can affect the input, processing, storage and output of information. The susceptibility of the software to error increases exponentially with the number of variables that are involved in its operation. Even if software is known to be "mature" and "stable," as the ED suggests, there are variables and values that the software designers and implementers may not have foreseen or taken action to mitigate that may adversely affect the reliability of the software even after a considerable amount of time. For example:

    a. In the late 1990's, many software vendors had to rewrite their software code in order to accommodate a potential high-risk situation in which software would treat the year 2000 as the year 1900. This scenario, known as "Y2K," affected mature and well understood complex software systems. Even software that had been used for several decades could not withstand a set of values that came before it because design limitations did not foresee certain values as possible.

    b. Software that was designed to calculate interest rates with a rounding of six decimal points used a certain memory format that allowed it only to operate properly within interest rates that did not exceed 10%. However, as the interest rate used exceeded the limit, the software, though it was considered mature and stable, would not operate as designed.

Software design is not implemented in a vacuum in terms of the hardware, operating system, and middle-ware that underlie its operation. Changes to operating systems, shared libraries from which the software applies certain common functions, and changes to drivers and hardware might affect the reliability of the software without advance warning.

The reason these unintended consequences occur is that there is an inherent risk in the design of complex software systems. That risk is rooted in the method by which most software is developed. Software is generally developed through iterations of higher level operations: machine languages are abstracted into programming languages; programming languages are abstracted to programming libraries of reusable code; and libraries of code and abstract algorithms are used as high-level tools to create complex software systems. With these multiple levels of abstraction, the inherent risk increases as each level's code developer assumes that the underlying levels will operate as designed. Many times, underlying levels, such as operating systems or shared libraries, do not operate as designed and seemingly innocent upgrades or updates may adversely affect the reliability of the software.

We are of the opinion that, with respect to sections 154 and 157–159, for the reasons indicated above, the benchmarking of software, including that considered to be "mature" and "stable," is inadvisable. Alternative procedures should include tests of the controls, regression analysis and parallel operations (re-performance). In applying such

procedures, auditors should be cognizant that the maturity and stability of software might provide a false sense of security due to the complexity of the variables, operations and values that the software contains.

Section 155 describes the reliability of program change control that should be evaluated in an entirely automated application. We agree that the risk of the reliability of a software is directly related to the quality of the "system life cycle and development" process, or program change controls.

Additional Comments

Conceptual Framework
The concept of applying a benchmarking method to a control simply because it is believed to be "fully automated" gives rise to some concerns. Attestation professionals might not often find a modern IT environment that is fully automated in substance. Further, applying a benchmarking method to a manual control would not comply with current attestation standards. To propose the allowance of an application of a benchmarking method of an automated control might be viewed as an internal contradiction in the applicable attestation standards.

External Factors
Benchmarking is an evaluation method that is most suitable to legacy EDP systems. Such systems are now rare because linear processing (input-process-output) is not frequently found in today's business environment. Linear processing has generally been replaced by complex IT environments. The marketplace's expectations are that attestation engagements will be able to address complexities that include ever-changing input definitions and processing parameters. It is our belief that, in situations in which a benchmarking method is applied and there is the occurrence of a material misstatement, the method might not satisfy external reviews or legal thresholds.

Conclusion

We believe that the proposal to allow a benchmarking method for a fully automated control is inappropriate given the state of information technologies and marketplace expectations. Instead, the standards should require tests of controls of fully automated systems in a manner similar to the testing and evaluation currently applied to non-automated controls by attestation professionals.