MANUEL BARBERO
345 GREENWICH STREET Apt. 3B - NEW YORK, NY 10013
+ 1 917 216-0782 ● mbarberony@yahoo.com

To whom it may concern,

As a technology executive practitioner with 30+ years of experience in technology consulting and operations in the Financial Service industry (see my public profile at www.linkedin.com/in/mbarberony) I welcome the opportunity to contribute to the agencies' work towards enhancing cyber risk management standards.

My comments pertain primarily to standards applicable in three of the five domains under the scope of the ANPR, namely cyber risk governance, management, and resilience building – all within the scope of software management.

Here are a few principles for your consideration which can guide the development of these standards:

- From a practical standpoint, any standard (including those promoted or mandated by governmental entities) tend to be more welcomed and to gain adoption in a faster and broader way when they also contribute to the target organizations' efficiency;
- Deploying cyber-risk standards can and should be considered within the broader scope of enhancing the overall quality of the technologies involved. Better code and infrastructure leads to safer and more resilient code and infrastructure[1];
- Fortunately, in the software realm, the technology community have had a long history of considering security as a key dimension of code quality that should be measured[2];
- That thinking has been particularly useful with respect to so-called integration risks, i.e. the risks that are created when several components of a system are coupled [3];

---

[1] *Cyber risk created through sloppily developed software: IT risk encompasses many human, process, and technical considerations. While appropriate processes and responsible people are essential, it is impossible to enforce best practices in a continuous and consistent manner. Technically, severe IT failures take root in three main areas – hardware, network, and software. While the hardware and network plumbing are increasingly reliable, real life experience has taught us that most major failures are now coming from the software layer. Bad software programming and software upgrade practices are the root cause of most of the catastrophic IT failures everyone can read in the press weekly. Worse yet, these public failures represent only a tiny part of the incidents handled regularly within IT organizations.*

[2] *Cyber Risk and Code Quality - Good programming practices have been a topic for decades now, usually falling under the "code quality" umbrella. There have been numerous code quality papers from tools vendors, academies and standardization organizations such as ISO, but these have always focused on the quality of the code within a particular program, to ensure cleanliness and proper execution of a particular subroutine. The focus of code quality is typically on proper syntax, readability, code hygiene and good basic practices. Code hygiene is not sufficient to address cyber risk.*

[3] *Integration Risks -- The rub here is that the exact same piece of code can be safe or highly dangerous, depending on the context in which it operates. In other words, an IT system can be made of thousands of programs of excellent code quality, and still be a complete disaster. This is a serious industry problem because most of the non-technical executives in IT still believe that if code quality is technically good, the system will be technically good. Numerous scientific and empirical studies (please see references below) have demonstrated this is not correct. The same way you would not say a brick building is structurally safe, resilient, and secure just because it has been built with high quality red bricks.*

- This is something that current software architecture paradigms (e.g. microservices, component, messaging architecture, etc.) make all the more important to consider, as many enterprise solutions are in fact an amalgamation of sub-systems which can be individually safe but vulnerable when aggregated.

In that context, I would like to bring the agencies' attention to the remarkable work that has been accomplished by a consortium founded by the Software Engineering Institute (SEI) at Carnegie Mellon University, and the Object Management Group (OMG), a neutral standardization organization, together with the MITRE Corporation and other federally funded research organizations. The name of this consortium is CISQ, standing for Consortium for IT Software Quality (www.it-cisq.org)[4].

This group has been able to define and promote an approach to manage and enhance software quality in a holistic way which explicitly includes security risk prevention in the definition of software quality (in a way that can be measured and audited). The CISQ approach is consistent with the above principles in that CISQ aims at fostering good design quality to create, not destroy value. It also puts some deserved emphasis on system-wide risk analysis.

Based on the above, and some of the additional references listed in the footer, and on common sense regarding the structural quality and integrity of complex, systems I strongly believe that the enhanced risk standard would do well to quantify software risks in terms of the CISQ specification.

Following these pragmatic technical rules and guidelines, which are public and free, software development organizations can quantify progress against industry norms, and deliver more robust, secure and safe IT systems.

Please feel free to contact me if you have any additional questions about my comments.

Regards,

Manuel Barbero

In software engineering, like in all other engineering matters, most of the severe software glitches are coming from poor assemblage, two good programs poorly interconnected forming a risky system. It is called quality of the architecture, or "system-level quality", coming as a natural complement of the far too simplistic "code quality" focus.

[4] *Existing CISQ Work -- The CISQ has been working the past 8 years to define a set of standards and technical recommendations on how to architect and code reliable, resilient and secure software systems. These include architectural and system-level considerations, considering how the programs interact between each other, and particularly how the programs access and manipulate data [See good architectural practice at system level in the main table, columns on the right: http://it-cisq.org/standards/].*

References

- The CISQ has its semi-annual Cyber-Resilience Summit in Washington DC (http://it-cisq.org/cyber-resilience-summit-2017/); during past summits Dr. Gilmore, Director of Operational Test and Evaluation for all DoD, stated that security and assurance needs to include structural quality for software, and acquisition needs to be based on standards such as CISQ.
- The SEC Regulation SCI (Systems Compliance and Integrity) has named the CISQ as a standard body to be considered.
- CEO of the Object Management Group (OMG), Dr. Richard Soley, IEEE and MIT fellow, has written a detailed paper on structural quality that includes numerous industry and scientific references: http://www.omg.org/CISQ_compliant_IT_Systemsv.4-3.pdf .
- Wikipedia highly recommend the use of CISQ rules in its main "software quality" page, which has been reviewed thousands of times and includes 38 references from industry and academia: https://en.wikipedia.org/wiki/Software_quality .