the newsletter of Reliability, Maintainability, and Supportability

June 2019
Volume 23
Issue 2

in this issue

Does Reliability Function Accurately Predict System Reliability? by Jezdimir Knezevic

Abstract

According to Knezevic [1] the purpose of the existence of any functional system is to do work. The work is done when the expected measurable function is performed through time. However, experience teaches us that expected work is frequently beset by failures, some of which result in hazardous consequences to: the users; the natural environment; the general population and businesses. During the last sixty years, Reliability Theory has been used to create failure predictions and try to identify where reductions in failures could be made throughout the life cycle phases of a functionable system type.[1] However, mathematically and scientifically speaking, the accuracy of these predictions, at best, were only ever valid to the time of occurrence of the first failure, which is far from satisfactory in the respect of its expected

1 According to Knezevic [1] functionable system type is "a set of mutually related entities put together to do a functionability work in accordance to physical laws and given functionability rules."

Continued on Page 2

Reliability Requirements Often Determined by Much More than Probability Metrics

by Russell A. Vacante, Ph.D.

It is often said that establishing reliability requirements are just as much of an art as based on sound scientific principles. Jezdimir Knezevic features an article in this newsletter, which provides a good discussion on how to improve the "science" of reliability beyond the use of probability metrics. My editorial discussion, on the other hand, discusses other influences that impact reliability requirements often to the determinant of system/mission objectives.

Adequate funding, which is needed to support the design, engineer, and life-cycle management support team and could be utilized to build robust system requirements, is often viewed by high-level decision makers as a costly, early-on endeavor, during the life-cycle (Concept Exploration and Engineering Development). Since reliability is essentially a life-cycle probability growth metric that can't demonstrate a short term return on investment, for the time experts spend doing mathematical calculations and related computer design modifications, it receives little support from upper management. These decision makers need to justify their expenditures to shareholders in ways that

Continued on Page 8

life. Consequently, the main objective of this paper is to raise the question how reliable are reliability predictions of functionable systems based on the Reliability Function and to introduce MIRCE Functionability Equation [2] to reliability practitioners as a way forward towards more accurate life long predictions, which would certainly enhance the reputation of their profession and closer collaboration with other well-respected and proved engineering disciplines.

1.0 Introduction

The necessity for the reduction in occurrences of operational failures started with the advanced developments of military, aviation and nuclear power industries, where the potential consequences could be significant. And so, during 1950s, Reliability Theory was "created." It was based on mathematical theorems rather than on scientific theories. Massive attempts where made to further the applications of the existing mathematical, statistical and analytical methods without a real understanding of the mechanisms that caused the occurrences of in-service/operational failures.

Not surprisingly, deterministically educated engineers and managers experienced fundamental difficulties in understanding Reliability Theory. The reason for that is very simple. Probability, unlike numerous measurable physical properties and as a main concept of reliability, cannot be seen or measured directly, for example: pressure: temperature: volume: weight of a component can be measured directly and by using appropriate mathematical manipulations, accurate predictions of the corresponding properties of a system constructed of these parts can be obtained. Moreover, the occurrence of a component failure is also clearly manifested and physically observed phenomena. And yet, the concept of reliability is abstract and immeasurable. It cannot be seen on the component/system. In fact, it serves as an abstract property of a component/system that obtains a physical meaning only when a large sample of components/systems is considered.

2.0 Reliability Function

To support the above presented conclusions regarding Reliability Theory, the fundamental definition of reliability will be used and analysed. It is widely accepted that Reliability is defined as the probability (*P*) that a considered entity (component, product, system) will operate without failure during a stated period of time (*t*), when operated in accordance with defined parameters. Mathematically, this statement is fully defined by the Reliability Function, *R(t)*.

2.1 Reliability Function of a Component

For any component considered, the reliability function is defined in the following manner:

$$R(t) = P(TTF > t) = \int_t^\infty f(t)dt, \qquad t \geq 0$$

where: *R(t)* is the reliability function, *f(t)* is the probability density function of the random variable known as the Time To Failure (*TTF*) of a component.

Reliability data regarding components can be fully defined through the numerous well-known probability distributions. However, in the vast majority of cases, current industry practices are premised on the reliability of components being defined by their manufacturers through a constant failure rate, λ, which forces all interested parties to express the reliability function in the form, $R(t) = e^{-\lambda t}$

2.2 Reliability Function of a System

The Reliability function for a system, $R_s(t)$, is determined by the reliability functions of the constituent components and the way they impact the failure of the system. For example, the reliability function for the system, whose reliability block diagram is presented in Figure 1, is fully defined by the following mathematical expression:

$$R_S(t) = P(TTF_S > t) = R_A(t) \times \{1 - [1 - R_B(t)][1 - R_C(t)]\}, \qquad t \geq 0$$

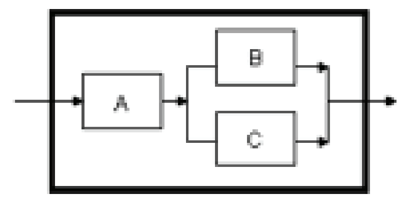


Figure 1: Reliability Block Diagram for a Hypothetical System whose failure will occur if a component A fails, or if components B and C fail.

The above two equations briefly summarise the essence of the reliability function when the main concern is a prediction of the behaviour of the system until the first failure.

3.0 Mathematical Reality of a Reliability Function

Being educated to use mathematical expressions for all engineering predictions, which always have a single numerical outcome, the author has spent over a decade understanding the fundamental physical meanings of the mathematical definitions for the reliability of systems by the system reliability function. Thus, the realisation was that reliability mathematics dictates the following physical reality of the systems considered:

- One Hundred percent quality of components production and installation

- Zero percent of transportation, storage and installation tasks
- One Hundred percent of components are mutually independent
- No maintenance activities (inspections, repair, cleaning, etc.)
- Continuous operation of the system (24/7)
- First observable failure is a failure of the system
- Time counts from the "birth" of the system
- Fixed operational scenario (load, stress, temperature, pressure, etc.)
- Operational behavior is independent of the location in space (GPS or stellar coordinates)
- Reliability is independent of humans (operators, users, maintainers, managers, general public, law makers, etc.)
- Reliability is independent of calendar time (seasons do not exist)

4.0 Physical Reality of Reliability Function

Systematic research performed by the author during several decades of the observable physical realities of in-service/operational life of aerospace, military and nuclear power industries have clearly shown that the flowing physical reality determines the reliability of systems [1].

- Quality of produced components and assemblies is less than 100%
- There are huge interactions between "independent" components
- Maintenance activities like: inspections, repair, cleaning, etc., have significant impact on the life of a system and impact reliability
- Neither all systems nor all components operate continuously (24/7)
- First observable failure is not necessary the failure of a system (failure of components B or C alone, in the Figure 1, does not cause system failure)
- Components and a system have different "times"
- Variable operation scenarios (load, stress, temperature, pressure, etc.)
- Reliability is dependent on the location in space defined by GPS coordinates
- Reliability is dependent on humans, like: users, maintainers, general public
- Reliability is dependent on calendar time

5.0 Closing Question

The above list of physically observed and undeniable facts seriously impact the accuracy of the reliability predictions currently provided through reliability theory. Because, the first failure event and all subsequent ones generate physically observable changes in the reliability of a system that are impossible to embrace by the existing concepts used in the formulation of the Reliability Function.

The closing question for all reliability professionals is, "How can predictions of functional system reliability be "reliable" when lifelong physically observable events and associated human rules are totally excluded from the predictions?"

6. Way Forward for Reliability Community

At the beginning of 20th century physicists had a problem to explain and predict the behaviour of subatomic particles, which totally "disobey" the traditional uniform motion well understood and mathematically defined by Newton, Maxwell and other scientists. Instead to modify the nature, they work hard to modify the mathematics and made it fit physically observed reality. Thus, the quantum mechanics has been created and successfully applied through nuclear engineering, ever since. Having found myself in the similar position at the end of 1990s regarding Reliability Theory, on one hand and physically observed reality, through car rallying, on the other, I spent nearly 20 years to develop a science based approach to reliability predictions of functionable systems and complementary mathematical scheme that enables accurate representation of physically observed reality with the language of mathematics. The new body of knowledge replaced the exiting reliability function that deals with the time to the single failure, with MIRCE Functionability Equation [2] that deals with all failures of a functionable system type, including adopted maintenance philosophy and support strategies to deal with each of them. The new body of knowledge was named MIRCE Science and is fully documented in [1]. ■

References

1. Knezevic, J., The Origin of MIRCE Science, pp. 232. MIRCE Science, Exeter, UK, 2017, ISBN 978-1-904848-06-6
2. Knezevic, J., Mirce Functionability Equation, Int. Journal of Engineering Research and Applications www.ijera.com ISSN : 2248-9622, Vol. 4, Issue 8(Version 7), August 2014, pp.93-100

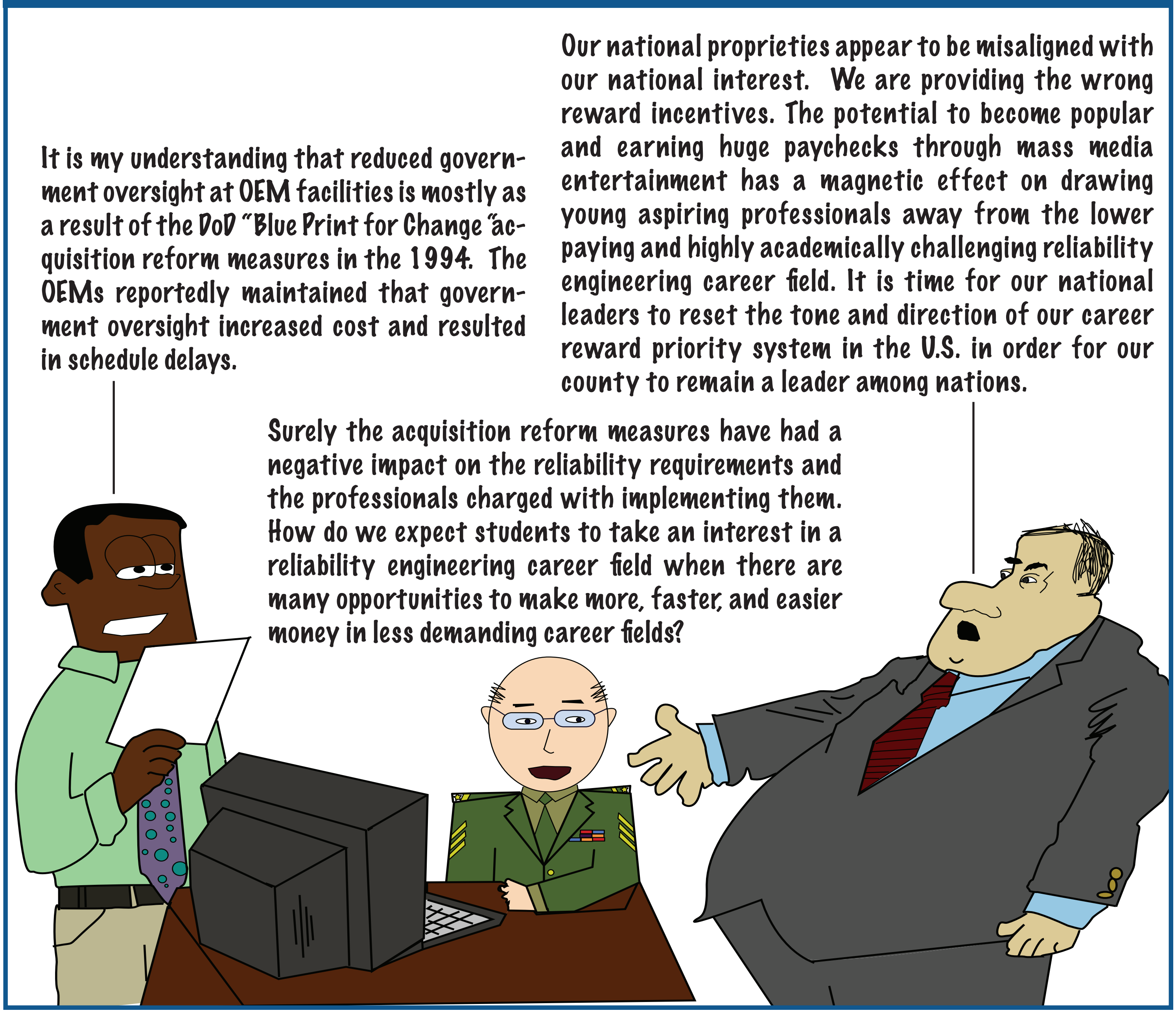
RMS Partnership Membership

Sign up today! Membership Dues Only $35.00 Annually.
See Membership Benefits and Registration at:
www.rmspartnership.org

Please contact president@rmspartnership.org
if you have questions about membership.

Another Day At The Office

by Russell A. Vacante, Ph.D.



THE BALUSTER GROUP

SERVICE DISABLED VETERAN-OWNED SMALL BUSINESS

Management Expertise • Engineering Experience • Strategic Insights

Management & Technical Expertise

The BALUSTER Group is a Service Disabled Veteran Owned Small Business (SDVOSB) that solves challenging management and technical questions. The company has competencies with regulatory, engineering and technical expertise, as well as, claims and litigation experience. BALUSTER's management and engineering consultants are experts in their fields and provide clients access to a full range of management, engineering, scientific and regulatory capabilities.

The company's core strength lies in its' management and technical personnel's comprehensive multi-disciplinary expertise and practical experience. With advanced degrees (Ph.Ds, MBAs, M.Eng, etc.), regulatory expertise and industry knowledge, BALUSTER's team is uniquely capable of addressing a wide range of projects, regardless of complexity or timeline.

The BALUSTER Group Team

The BALUSTER Group's Federal team offers a unique collaborative approach to solving problems that focus on appropriately staffing each engagement with the best-suited subject matter experts. This allows our clients to leverage the depth of knowledge held across our management and technical practice groups.

The Federal team is comprised of former senior executives from the U.S. government, business management executives, senior level engineers, policy analysts, and researchers.

COMPANY CAPABILITIES

- Accident Investigation & Analysis
- Aircraft Accident Reconstruction
- Applied Mechanics
- Automotive Accident Reconstruction
- Automotive Defect Analysis
- Automotive Regulatory Compliance
- Automotive Research & Testing
- Autonomous Vehicle Technology Integration & Policy
- Aviation Operations and Policy
- Block Chain Integration
- Chemical Engineering & Analysis
- Child Product Safety
- Commercial Vehicle Operation, Safety and Regulations
- Consumer Product Safety
- Crash Avoidance Technologies
- Crashworthiness
- Design Analysis
- Energy Analysis & Integration
- **Engineering Consulting**
- **Environmental Engineering & Science**
- Fire & Explosion Investigation & Analysis
- Hazardous Material & Waste Transportation
- Heavy Trucking & Commercial Vehicles
- Industrial Hygiene & Incidents
- Infrastructure
- **Management Consulting**
- Mechanical Engineering
- National Environmental Policy Act (NEPA)
- Railroad Technology & Safety
- Regulatory & Compliance
- Research & Design
- Risk Assessment & Analysis
- Safety Analysis & Engineering
- Statistics & Reliability
- Transportation Engineering & Research

www.theBALUSTERgroup.com

To learn more about the BALUSTER Group Federal team, call at (571) 334-4807, or send an email to cjbonanti@theBALUSTERgroup.com

SERVICE DISABLED VETERAN-OWNED SMALL BUSINESS

Is Hardware Really That Much Different From Software

by John Blyler

When can hardware be considered as software? Are software flows less complex? Why are hardware tools less up-to-date? Experts from ARM, Jama Software and Imec propose the answers.

The Internet-of-Things will bring hardware and software designers into closer collaboration than ever before. Understanding the working differences between both technical domains in terms of design approaches and terminology will be the first step in harmonizing the relationships between these occasionally contentious camps. What are these differences in hardware and software design approaches? To answer that question, I talked with the technical experts including Harmke De Groot, Program Director Ultra-Low Power Technologies at Imec; Jonathan Austin, Senior Software Engineer at ARM; and Eric Nguyen, Director of Business Intelligence at Jama Software. What follows is a portion of their responses. — JB

***Blyler:* The Internet-of-Things (IoT) will bring a greater mix of both HW and SW IP issues to systems developers. But hardware and software developers use the same words to mean different things. What do you see as the real differences between hardware and software IP?**

De Groot: Hardware IP, and with that I include very low level software, is usually optimized for different categories of devices, i.e. devices on small batteries or harvesters, medium size batteries like mobile phones and laptops and connected to the mains. Software IP, especially for the higher layers, i.e. middleware and up can easier be developed to scale and fit many platforms with less adaptation. However, practice learns that scaling for IoT of software also has its limitations, for very resource limited devices special measures have to be taken. For example direct retrieval of data from the cloud and combining this with local sensor data by a very small sensor node is a partly unsolved challenge today. For mobiles, laptops and more performing devices there are reasonable solutions (though also not perfect yet) to retrieve cloud data and combine this with the sensor information from the device in real-time. For sensoric devices with more resource constraints working on smaller batteries this is not so easy, especially not with heterogeneous networking challenges. Sending data to the cloud (potentially via a gateway device as a mobile phone, laptop or special router) seems to work reasonably, but retrieving the right data from the cloud to combine with the sensor data of the small sensor node itself for real-time use is a challenge to be solved.

Austin: Personally, I see three significant differences between the real differences between hardware and software design and tools:

1. How hard it is to change something when you get it wrong? It is “really hard” for hardware, and somewhere on the spectrum from “really hard” to “completely trivial” in software.
2. The tradeoffs around adding abstraction to help deal with complexity. Software is typically able to ‘absorb’ more of this overhead than hardware. Also, in software it is far easier to only optimize the fast path. In fact, there usually isn’t as much impact to an unoptimised slow path (as would be the case in hardware.)
3. There are differences in the tool sets. This was an interesting part of an ongoing debate with my colleagues. We couldn’t quite get to the bottom of why it is so common for hardware projects to stick with really old tools for so long. Some possible ideas included:

- The (hardware) flow is more complex, so getting something that works well takes longer, requires more investment and results in a higher cost to switch tools.
- There’s far less competition in the hardware design space so things aren’t pushed as much. This point is compounded by the one above, but the two sort of play together to slow things down.

- The tools are harder to write and more complex to use. This was contentious, but I think on balance, some of the simplicity and elegance available in software comes because people solve some really touchy physical issues in the hardware tools.

So, this sort of thinking led me to an analogy of considering hardware to be very low level software. We could have a similar debate about javascript productivity versus C – and I think the arguments on either side would be quite similar to the software versus hardware arguments.

Finally on tools, I think it might be significant that the tools for building hardware are *software* tools, and the tools for building software are *also* software tools. If a tool for building software (say a compiler) is broken, or poor in some way, the software engineer feels able to fix it. If a hardware tool is broken in some way, the hardware engineer is less likely to feel like it is easy to just switch tasks quickly and fix it. So that is I guess to say, software tools are built for software engineers by software engineers, and hardware tools are built by software engineers to be sold to companies, to be given to hardware engineers!

Nguyen: One of the historical differences relates to the way integrated system companies organized their teams. As marketing requirements came in, the systems engineers in the hardware group would lay out the overall design. Most of the required features and functionality were very electrical and mechanical in nature, where software was limited to drivers and firmware for embedded electronics.

Today, software plays a much bigger role than hardware and many large companies have difficulties incorporating this new mindset. Software teams move at a much faster pace than hardware. On the other hand, software teams have a hard time integrating with the tool sets, processes and methodologies of the hardware teams. From a management perspective, the "hardware first" paradigm has been flipped. Now it is a more of software driven design process where the main question is how much of the initial requirements can be accomplished in software. The hardware is then seen as the enabler for the overall (end-user) experience. For example, consider Google's Nest Thermostat. It was designed as a software experience with the hardware brought in later.

Blyler: Thank you. ■

Originally published in "System Design Engineering" by the author.

Interested In Contributing?

If you are interested in sharing your knowledge in future editions, please contact Russ Vacante at president@rmspartnership.org

Articles can range from one page to five pages and should be of general interest to our members.

Editorial, from Page 1

increase their investment wealth.

Next there are political considerations. Original Equipment Manufactures (OEMs) don't want to be told by anyone, especially government oversight officials, how to design, develop, and build their systems. The acquisition reform measure initiated by Secretary William J. Perry, known as the "Blue Print for Change" in 1994 resulted mostly from industry pressure to weaken government regulations and oversight since they presumably were a cost driver to industry (where have all the saving from this cost avoidance initiative gone?).

In the years subsequent to acquisition reform we have seen an erosion of professional reliability career fields, along with a substantial decline of well-trained reliability professionals in both industry and government engineering organizations. Proverbially speaking, there are very few reliability design professionals and watchdogs ensuring that speed of delivery due to competition and budget cuts don't adversely impact reliability design requirements. While it cannot be definitely stated that politicians who require funds to get reelected are influenced by their industry contributors' interests to reduce government regulations and oversight it, conversely, cannot be said that their support for acquisition reform is free from such influence.

From a cultural-economic perspective we have a host of intertwining and complex issues. The first is that society's economic priorities seem to be out of whack. Students who are U.S. citizens are not clamoring to enter challenging engineering and science professions. Instead, they are opting for majors in business management, computer programming and gaming, and performance courses that mostly pay better and have greater job security than the more academically challenging subjects of mathematics and science. In addition, to mention another current job trend among many millennials (Generation Y), is their focus on becoming famous through mass media self-promotion. In contrast, we frequently find occupying the engineering and scientific classrooms of universities and colleges a large foreign student population. These students, after successfully completing their studies in the U.S., often return home to well paying, stable, and high status jobs. They are employed in professional positions designed to meet the technical competitive market and defense needs of their nation.

A second cultural phenomena associated with the U.S. job market pertains to economic opportunity. Within industry itself, probably due in part to a decline in the number of technical experts, U.S. employees frequently change jobs in search of improved wages and work environments. Similarly, the wages of most government organizations cannot compete with the private market place. So after gaining some limited work experience government employees, speaking primarily of Gen X folks, seek and acquire higher paying industry jobs. Consequently, few Generation X technical folks are successfully groomed and mentored to become seasoned professionals in reliability and related fields that can be relied upon to provide meaningful industry oversight and assistance and/or become subject matter experts in industry.

Lastly, the declining support among industry leaders making reliability requirements an integral part of the life-cycle results in few resources provided for training and general support for folks who are tasked with reliability job responsibilities. Consequently, the number of reliability experts continues to decline given that few opportunities exist for career growth that pertains to ensuring that reliability is an integral part of system life-cycle design. This circumstance results in a self-fulfilling prophecy. Fewer native-born students enroll in technical courses in our colleges and universities that, in turn, results in fewer reliability professional employed in critical engineering positions in government and industry.

If there is a moral to this discussion it is that the time has come for government, industry and individuals to work towards a greater good—something that is more important and larger than immediate profits, career growth and instant fame. We are dependent upon reliable, safe systems for defense, domestic use, and international commerce and communications as advance technology evolves and proliferates at a record setting pace. To ensure reliability is maintained as an integral part of the systems engineering life-cycle process, scientific metrics have to be implemented, as well as, outside influences that go beyond pure engineering principles be eliminated. ■