Requirements Engineering for Safety-Critical Systems

Overview and Challenges

Luiz Eduardo G. Martins, Federal University of São Paulo

Tony Gorschek, Blekinge Institute of Technology

// Requirements engineering is crucial to developing and maintaining safety-critical systems. Researchers studied the literature and interviewed practitioners to discover what approaches are available for capturing, specifying, and communicating safety requirements and to determine the remaining challenges. //

EXPERIENCE REPORTS ON safety-critical systems (SCSs) show many cases in which systems failed because of insufficient requirements specifications or misunderstandings traced to problems in requirements engineering. Such failures have contributed to accidents that have harmed the environment, property, and people, including injury and even death. Accidents have a strong negative impact on the image of the companies responsible for the associated systems. Almost all accidents with serious consequences in which software was involved can be traced to requirements failures, and particularly to incomplete requirements.[1]

So, requirements specifications and the related requirements-engineering processes play an important role during safety certification of SCSs.[2] With SCSs' increasing complexity, the rules and standards for safety certification and the associated processes defined by governments and international agencies are becoming more difficult and extensive. In addition, with system functionalities increasingly moving from hardware to software, safety certification is becoming even more complex.

As a response, the past four decades have seen significant research in improving SCS engineering. One of the most important challenges for companies that develop SCSs is to create and establish a complete, correct, unambiguous, testable, and yet understandable requirements specification.[3] A shared understanding among the stakeholders throughout the SCS lifecycle (including customers, system engineers, requirements engineers, safety engineers, developers, testers, external regulators, and suppliers) is critical.

We wanted to see whether the efforts to improve SCS engineering have paid off. So, we conducted a systematic literature review of 165 papers comprising industry experience reports and descriptions of new models.[4] The papers reported approaches to elicit, analyze, model, specify, and validate requirements for SCSs. They came from the ACM Digital Library, IEEE Xplore, SpringerLink, and ScienceDirect, covering 1983 to 2014. We also interviewed 18 experienced

0740-7459/17/$33.00 © 2017 IEEE

TABLE 1

An overview of the interviewed practitioners.

Participant	Domain	Education	Current role	Previous role	Experience with safety-critical systems (yrs.)
A	Defense & aerospace	PhD in engineering physics	Head of strategy	Software engineer	24
B	Defense & aerospace	Master's in computer science	System engineer	Software engineer	15
C	Defense & aerospace	PhD in human factors	Safety engineer	Human-factors specialist	13
D	Automotive	Bachelor of marketing	Functional-safety manager	Operational-development team leader	4
E	Automotive	Bachelor of mechatronics	Test leader in software	Software tester	2
F	Automotive	Bachelor of computer science	Safety leader	Control engineer	7
G	Medical devices	Master's in electronic engineering	System engineer leader	System engineer	8
H	Defense & aerospace	Master's in avionics	Aerospace project analyst	Quality trainee	4
I	Defense & aerospace	Master's in system engineering	System engineer	Business process analyst	5
J	Automotive	PhD in software engineering	Safety engineer	Application engineer	2
K	Industrial machinery	Bachelor of electrical engineering	Safety advisor	Electrical designer	>10
L	Industrial machinery	PhD in software engineering	Requirements engineer	Requirements engineer	4
M	Industrial machinery	Technician in electrical engineering	Safety leader	Safety officer	>14
N	Industrial machinery	Bachelor of graphical technology	Equipment safety officer	Safety officer	25
O	Industrial machinery	Bachelor of automation	Automation engineer	Automation engineer	16
P	Industrial machinery	Bachelor of metallurgy	Metallurgical sales engineer	Metallurgical sales engineer	7
Q	Telecommunications	Technician in electrical engineering	Product manager	Project manager	10
R	Maritime	Master's in electrical engineering	Project manager	Software engineer	20

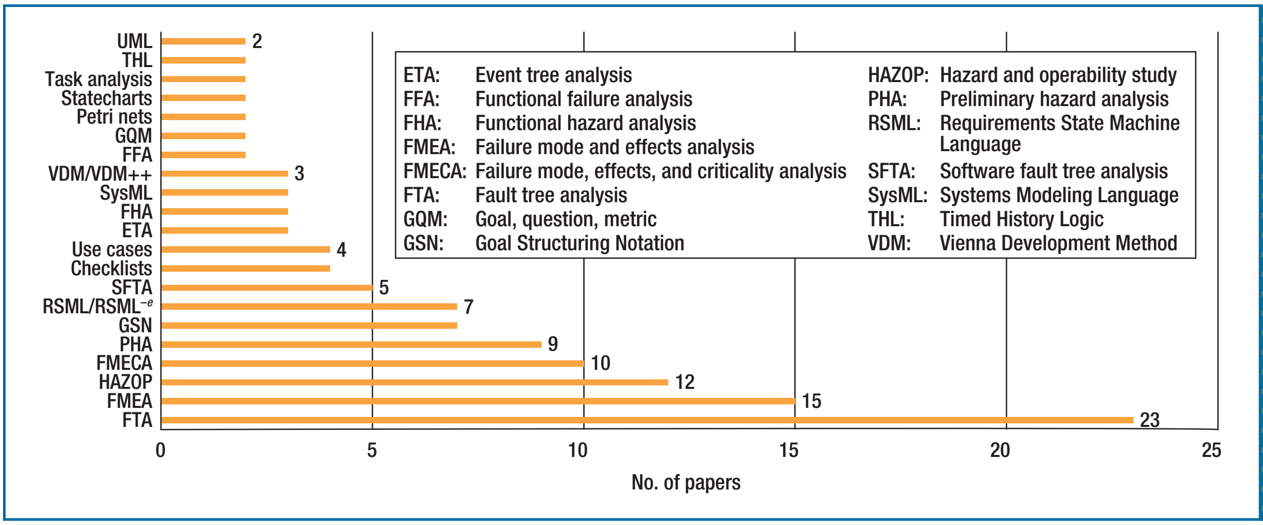

FIGURE 1. The most-cited approaches for capturing and handling safety requirements.[4] Practitioners largely preferred the traditional approaches.

practitioners from 10 SCS companies (see Table 1) to ascertain the approaches they used day-to-day and their perceptions of the approaches reported in the 165 papers.

The Traditional Approaches Still Prevail

Our literature review revealed that practitioners largely preferred the traditional approaches for capturing and handling both general and safety requirements.[4] (Figure 1 lists the techniques, methods, models, and languages reported.) The most adopted approach was FTA (fault tree analysis; 15 percent of the papers), followed by FMEA (failure mode and effects analysis; 10 percent), HAZOPs (hazard and operability studies; 8 percent), FMECA (failure mode, effects, and criticality analysis; 7 percent), and PHA (preliminary hazard analysis; 6 percent). (For more on FTA, FMEA, and HAZOPs, see the related sidebar.) These approaches are traditional in safety engineering.[5,6] FTA was the most frequent approach for safety analysis and also supported requirements elicitation, specification, and validation.

On the other hand, 5 percent of the papers reported newer approaches—for example, GSN (Goal Structuring Notation) and RSML (Requirements State Machine Language) or RSML^{-e} (RSML without Events). GSN was the only approach reported for documenting safety cases. We counted as UML only the papers describing research that used UML diagrams in an integrated way.

More interestingly, we found 73 unique approaches; that is, each one appeared in only one paper. This seems to indicate that there were several well-researched and well-developed approaches and many smaller approaches that weren't used to any significant extent. Many of the smaller approaches didn't undergo refinement; they were presented as one-off solutions in one case and weren't heard of again.

Our interviewed practitioners had the same tendency to prefer the traditional approaches. One explanation could be that most companies in the SCS context must follow standards and regulations. As one practitioner stated,

> *We have a set of international specific standards we have to follow to build the machines. We really need to identify what parts of the standards are applicable for the machinery we are designing (cutting, laminator, printing machine, and so on).*

The international standards that practitioners follow strongly recommend traditional approaches for safety analysis. For example, ISO 15998:2008, a safety standard for machine control systems (MCSs) using electronic components, states,

> *A documented analysis shall be included, indicating the realization of the safety concept as described. This may be done by an analysis (for example, FMEA, FTA, ETA) or using equivalent methods suitable for the safety concept of the MCS.*[7]

FTA, FMEA, AND HAZOPS

Fault tree analysis (FTA) is a deductive method for analyzing failure events that can put a system in an undesired state (for example, a hazardous state that can lead to an accident).[1] H.A. Watson developed FTA in 1962 at Bell Laboratories. It was originally used to evaluate an intercontinental-ballistic-missile launch control system. The tree is written using logic gate symbols (AND/OR).

Failure mode and effects analysis (FMEA) is an inductive method that supports safety engineers during system failure analysis.[2] FMEA was developed in the late 1950s to help engineers analyze military-system malfunctions. FMEA takes into account a review of components and subsystems to identify failure modes and their causes and effects. For each component or subsystem, the failure modes and their effects on the rest of the system are recorded on an FMEA worksheet.

Hazard and operability studies (HAZOPs) evaluate processes to identify risks or hazards to personnel or equipment and to prevent inefficient operation.[3] HAZOPs were developed in the 1960s in the context of chemical companies.

References

1. *Fault Tree Handbook*, US Nuclear Regulatory Commission, 2014.
2. R.J. Mikulak, R. McDermott, and M. Beauregard, *The Basics of FMEA*, 2nd ed., CRC, 2008.
3. F. Crawley and B. Tyler, *HAZOP: Guide to the Best Practice*, 3rd ed., Elsevier, 2015.

ISO 13849-1:2015, a machinery safety standard, states,

> *For the estimation of Diagnostic Coverage, in most cases, failure mode and effects analysis (FMEA, see IEC 60812) or similar methods can be used. In this case, all relevant faults and/or failure modes should be considered.*[8]

Thus, a clear message to researchers looking to get their "new and improved" methods into practice is that they should ensure and show compliance with the safety standards. For practitioners, the safe bet might be to apply the cited approaches.

The interviewed practitioners most frequently cited FMEA, FTA, and risk assessment.[9] However, unlike in the empirical studies, the most mentioned approach was FMEA (66 percent), not FTA (33 percent). A possible reason for this preference might be that the practitioners considered FMEA to be more practical. One practitioner said,

> *One part of the result of the FMEA would be ways of prevention, which are requirements in themselves.*

Another practitioner stated,

> *We take the requirements into account and assemble people with different experience to do the risk analysis—that is, the traditional risk assessment. We make the FMEA on the requirements and try to find those that impact the safety of the system or product.*

Just because the traditional approaches still prevail doesn't mean that they adequately address practitioners' needs. Rather, better choices might not be available, at least from the practitioners' viewpoint. Similarly, Nancy Leveson pointed out the need to move on from the traditional event-based accident-modeling approaches, such as FTA and FMEA, to new system-theoretic approaches.[1] System-theoretic approaches have the clear benefit of focusing on systems as a whole, following the assumption that some system properties can be treated properly only in their entirety.

Usability and Usefulness

According to the *technology acceptance model*, the two most important factors in technology acceptance are the perceived ease of use (usability) and usefulness.[10,11] Although researchers have proposed several approaches to capturing and handling safety requirements, evidence is lacking of how usable, useful, and adaptable these approaches are for practitioners.

In the papers in our literature review that reported new approaches, evidence of usability and usefulness was scarce—or superficial. Only 4 percent of the papers revealed evidence of usability, and only 1.5 percent had any evidence of usability that could actually be considered evidence. The situation was even worse in the papers that proposed tools to support techniques, models, or methods. For example, in the 29 papers that proposed tools to improve safety analysis, requirements specification, or requirements validation, only 2 percent presented medium or strong evidence of how the researchers measured the tools' usability. Regarding usefulness, the situation was a little better; 25 percent of the papers showed strong evidence of how to measure usefulness (61 percent showed medium evidence).

A balance between safety requirements approaches' usability and usefulness must be pursued. Approaches that are admittedly useful but difficult to employ and to adapt to the daily life of industry professionals are of little practical use. Unfortunately, the weak evidence of most new approaches' usability and usefulness hampers practitioners' decisions on which of them to use.

Reducing the Gap between Academia and Industry

The adoption of a new technology (method, technique, process, or tool) involves costs and risks. *Rigor* and *relevance* are essential measures that help practitioners make a decision, weighing in the level of trust that can be put in the new technology.

To assess how new approaches for handling safety requirements are validated by their proponents, we applied Martin Ivarsson and Tony Gorschek's model for evaluating the rigor and industrial relevance of technology evaluations.[12] The model enables a classification in order to characterize research performed in an applied field.

The model evaluates rigor on the degree to which

- the context is described,
- the study design is described, and
- the validity is discussed.

On the basis of these three aspects, the level of rigor can range from 0 to 3.

The model evaluates industrial relevance on the basis of

- the subjects who participated in the studies,
- the context in which the studies were performed,
- the scale used in the studies' evaluation, and
- the research method adopted in the studies.

The first three aspects are concerned with the realism of the environment in which the studies took place. The fourth aspect is concerned with how the research method influenced the results. On the basis of these aspects, the level of relevance can range from 0 to 4.

We applied the rigor-and-relevance model to 151 of the studied papers. (We excluded 14 of the original 165 papers because they didn't adequately apply rigor and relevance.) Figure 2 gives details regarding the subjects, context, scale, and research methods reported in the papers. In 58 percent, only researchers participated in validating the approaches. In 27 percent, researchers and practitioners collaborated, and in 15 percent, only practitioners participated.

Figure 3 shows the results. Regarding rigor, 25 percent of the papers had a score equal to or greater than 2; 48 percent had a score of less than 1.5. Regarding relevance, 31 percent of the papers had the highest score (4 points); 38 percent had the lowest score (0 points).

These results are good compared to the requirements-engineering field in general, indicating that SCS is a relatively mature domain. However, in an applied field such as requirements engineering,[13] research should enable (or least prepare for) technology transfer to industry. For this to become a reality, researchers from academia and practitioners from industry must work jointly in relevant research projects, particularly to validate new approaches.[13]

But when we analyzed how researchers validated the new safety requirements approaches, we found that most cases had no industry participation. This situation must change. As one practitioner said,

> *It is very important for us that new tools are tested out in our reality; otherwise, we do not trust it.*

Also, most validation of new approaches used down-scaled scenarios. However, such approaches must be scalable. As another practitioner said,

> *We have to know it will work with all our requirements and over time as the database grows.*

Communicating Requirements

Almost all the papers we studied reported approaches that supported both safety engineers and requirements engineers. The approaches helped both of them in different but complementary activities, such as safety analysis and safety requirements analysis and specification.

This finding seems to suggest a tendency of integration between safety engineering and requirements engineering, shortening the gap between these disciplines.[3] This is a good thing; organizations need one integrated way of working, not a separation between safety engineering and requirements engineering. As one practitioner stated,

> *We need to work closely with developers and understand the customer needs at the same time as being safety aware.*

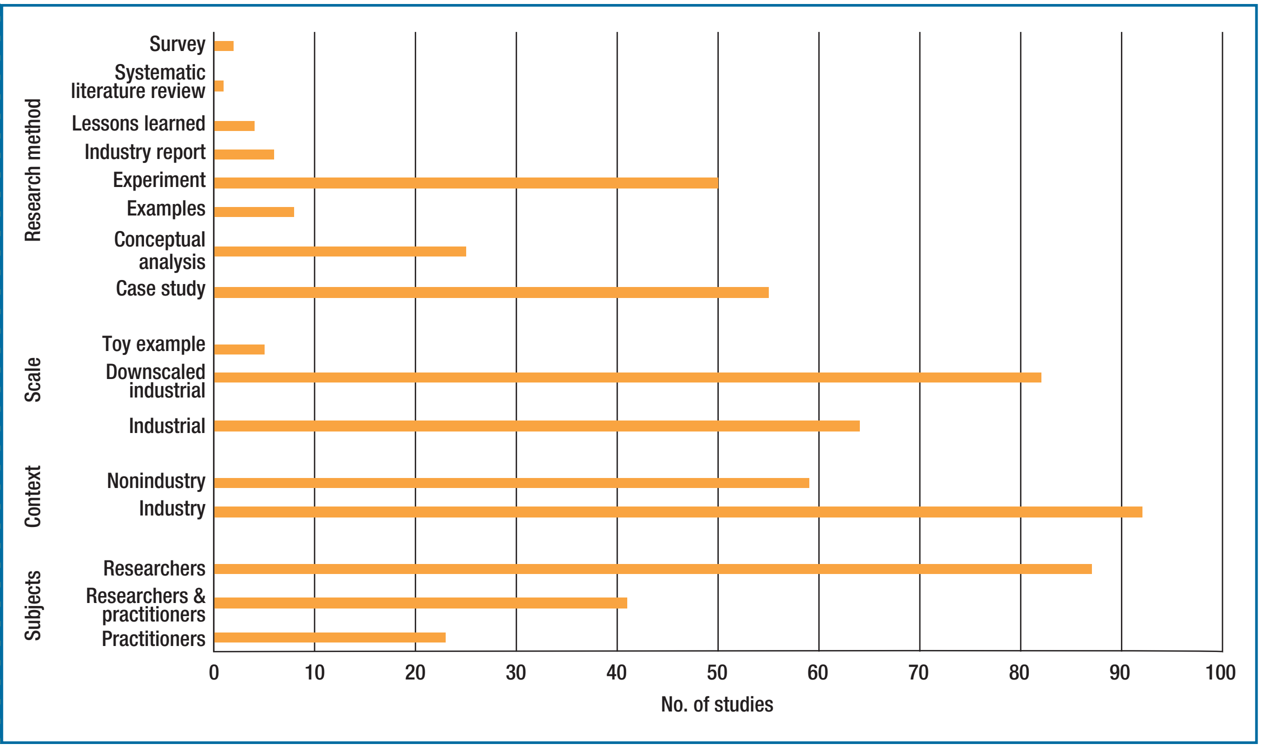

FIGURE 2. The subjects, contexts, scales, and research methods of the 151 papers we analyzed.

Few papers reported experimentation involving the other actors in SCS development and certification. For example, only three papers reported including certification auditors during validation—which might be an issue, because certification auditors significantly influence any approach being used.

Efficient communication is important throughout SCS development. That is, an approach should enable and support coordination and communication between different parties (for example, safety engineers, requirements engineers, developers, and auditors). Surprisingly, few of the papers addressed communication efficiency throughout the system-engineering lifecycle.[14]

Even more surprisingly, the interviewed practitioners reported the same situation. Most communication was informal. This isn't necessarily a problem; in agile teams, informality is often seen as, and can be, positive. However, it can often lead to missing or inadequate specifications, which is a critical problem in SCS development and certification.

Also, requirements analysis and safety analysis didn't inherently involve coordination or communication. In fact, the opposite was the case; these tasks often involved different tools and approaches used by different groups of engineers. As one practitioner said,

> *The communication between safety engineers and requirements engineers is good but very slow. This hinders communication about items that people deal with.*

As Table 1 shows, the practitioners were from many backgrounds and domains. But what's important is that most of them had at least 7 years' experience in SCS development. Also, approximately 50 percent of them played a leadership role in their company. Someone might argue that more-junior team members could be too focused on solving a specific task to worry (or know) about communication and coordination; this obviously wasn't the case in our study.

A vast majority of the practitioners recognized the need to improve communication and coordination and the integration of and collaboration among different roles and responsibilities. The current approaches didn't support this. As one practitioner said,

> *The teams are organized as silos; the safety team does "safety," the*

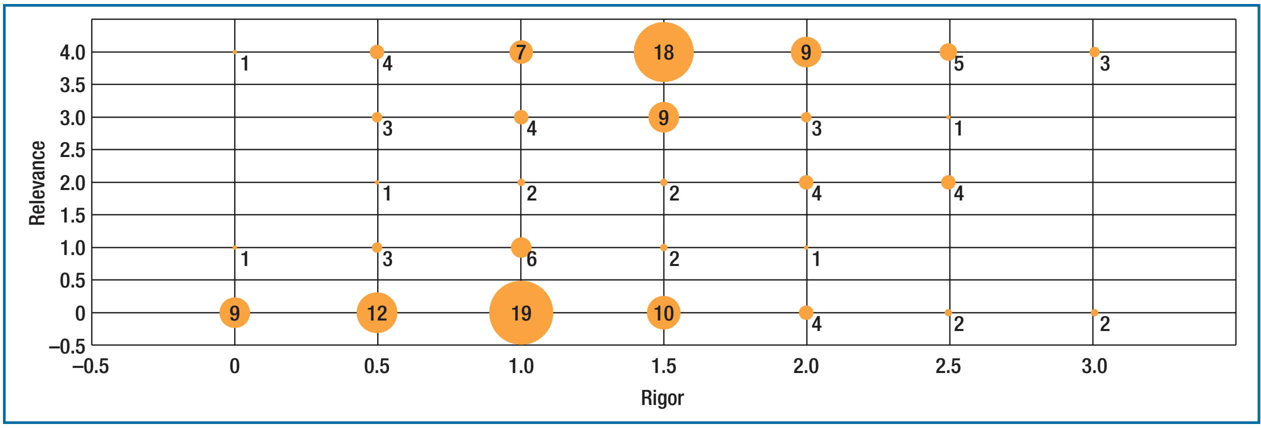

FIGURE 3. The rigor and relevance of the papers we analyzed (the numbers on the circles indicate the number of papers). The results are good compared to the requirements-engineering field in general, indicating that the safety-critical-systems domain is relatively mature.

> *requirements team does "requirements," and the software team does "software."*

One practitioner from the industrial-machinery domain reported on an attempt to coordinate and join more traditional requirements specifications with safety-critical aspects. The teams involved didn't write their own requirements. Rather, they used references to the safety standards, reusing the text and explanations from the standards. As the practitioner explained,

> *We have a lot of internal documents that document those requirements that we have set, as well as a lot of references to external standards. We are quite well documented in relation to what kind of requirements we put on the suppliers. We don't want to invent or rewrite what is available in the standards and already specifies exactly what we need. We try to keep our own requirements as few as possible, avoiding overlapping with the standards we adopt.*

This way of working can save time during specification. However, potential threats exist. The diverse readers of the requirements might have different understandings and a limited grasp of the key items elaborated only in the standards, which tend to use specialized language. This is especially true for non-safety engineers. This is further complicated in that they also weren't a part of the analysis or specification, so knowledge and understanding weren't transferred through the "doing" process. Also, a distributed specification, which points to information in another source, might have limited usability.

Rethinking Safety-Critical Systems

Over the past three decades, SCSs have been accepted and viewed as systems "whose malfunction can lead to accidents putting people, environment, property and mission in serious risk, including environmental catastrophes and loss of lives."[6] So, companies that produce aircraft, railway systems, and medical devices are obviously SCS companies. But what about companies that offer systems traditionally not seen as SCSs but on which people increasingly depend? One interviewed practitioner from a telecommunications company that offers mobile-communication systems and infrastructure didn't feel he was working in an SCS domain. However, as people increasingly use only mobile phones instead of also having landlines, shouldn't the base infrastructure that enables, for example, a call for an ambulance be considered safety critical?

Also, many companies in our study that develop traditional hardcore SCSs also develop other components, parts, products, and services—all integrated into one offering. This is the case in the automotive domain, in which the line between SCSs and normal features is blurring—take self-parking, for example. Normally, software features visible to drivers and passengers (such as for entertainment, comfort, or navigation) have been non-safety-critical. But with the advent of safety-critical software-based features that are exposed to (and to some extent controlled by) users, critical new challenges arise.

AN AGENDA FOR THE COMING YEARS

Regarding the state of the art and of the practice regarding requirements engineering for safety-critical systems (SCSs; see the main article), we propose this research agenda:

- To what extent does the combination of traditional and new approaches improve requirements communication among practitioners in SCS companies?
- What difficulties and barriers do industry practitioners face when changing from traditional event-based accident-modeling approaches to new ones, such as those based on the system-theoretic view of causality?
- What are the common problems of and conflicts between safety requirements approaches and security requirements approaches? How can we use this knowledge to create a common groundwork for developing new integrated approaches?
- To what extent do current safety standards help practitioners improve the safety requirements in multidomain systems?
- To what extent can lean and agile requirements-engineering approaches improve the integration of safety, requirements, test, and certification teams?
- To what extent can model-driven approaches assist requirements communication throughout the SCS lifecycle?
- How can we improve requirements-engineering education to better address the issues in the SCS context?
- To what extent do the current practices of requirements specification based on the safety standards address the communication issues between developers and suppliers?
- How have international standards organizations evaluated the new approaches to safety or hazard analysis?
- Do we need to rethink our concepts of SCSs, taking into account the new technologies that support the infrastructure of modern society? How will new concepts of SCSs affect system development, as well as safety standards and regulations?

Here, the collaboration of and coordination between SCS engineers and other engineers—through at least a shared understanding of requirements—is central to product success.

The current perception is that SCSs must directly control a safety-critical aspect. Given the overall evolution of products and services and the interdependence of systems, this perception might be inadequate or at least somewhat outdated. The good news is that we don't have to be restrictive. Exploiting employees' collective experience and using safety analysis, standards, and practices can greatly assist companies developing products traditionally not under the SCS umbrella. This would also help motivate collaboration between what's considered traditional requirements engineering and safety engineering—inspiring and motivating the development of new, integrated approaches for working, analysis, and specification.

We hope this brief discussion encourages researchers and practitioners to engage in more empirical and industry-oriented cooperation to create approaches and initiatives, and especially to evaluate new approaches. For our suggested research agenda, see the sidebar, "An Agenda for the Coming Years."

References

1. N.G. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*, MIT Press, 2011.
2. P. Rodríguez-Dapena, "Software Safety Certification? A Multidomain Problem," *IEEE Software*, vol. 16, no. 4, 1999, pp. 31–38.
3. J. Hatcliff et al., "Certifiably Safe Software-Dependent Systems: Challenges and Directions," *Proc. Future of Software Eng.* (FOSE 14), 2014, pp. 182–200.
4. L.E.G. Martins and T. Gorschek, "Requirements Engineering for Safety-Critical Systems: A Systematic Literature Review," *Information and Software Technology*, July 2016, pp. 71–89.
5. Y. Hiraoka et al., "Method of Computer-Aided Fault Tree Analysis and Safety Design," *IEEE Trans. Reliability*, vol. 65, no. 2, 2016, pp. 687–703.
6. N.G. Leveson, *Safeware: System Safety and Computers*, Addison-Wesley, 1995.
7. *ISO 15998:2008: Earth-Moving Machinery—Machine-Control Systems (MCS) Using Electronic Components—Performance Criteria*

and Tests for Functional Safety, Int'l Org. for Standardization, Apr. 2008; www.iso.org/standard/28559.html.

8. *ISO 13849-1:2015: Safety of Machinery—Safety-Related Parts of Control Systems—Part 1: General Principles for Design*, Int'l Org. for Standardization, Dec. 2015; www.iso.org/standard/69883.html.
9. B. Gallina, E. Sefer, and A. Refsdal, "Towards Safety Risk Assessment of Socio-technical Systems via Failure Logic Analysis," *Proc. IEEE Int'l Symp. Software Reliability Eng. Workshops*, 2014, pp. 287–292.
10. F.D. Davis, "User Acceptance of Information Technology: System Characteristics, User Perceptions, and Behavioral Impacts," *Int'l J. Man-Machine Studies*, vol. 38, no. 3, 1993, pp. 475–487.
11. P. Legris, J. Ingham, and P. Collerette, "Why Do People Use Information Technology? A Critical Review of the Technology Acceptance Model," *Information & Management*, vol. 40, no. 3, 2003, pp. 191–204.
12. M. Ivarsson and T. Gorschek, "A Method for Evaluating Rigor and Industrial Relevance of Technology Evaluations," *Empirical Software Eng.*, vol. 16, no. 3, 2010, pp. 365–395.
13. M. Ivarsson and T. Gorschek, "Technology Transfer Decision Support in Requirements Engineering Research: A Systematic Review of REj," *Requirements Eng. J.*, vol. 14, no. 3, 2009, pp. 155–175.
14. P. Ayrault, T. Hardin, and F. Pessaux, "Development Life-Cycle of Critical Software under FoCaL," *Electronic Notes in Theoretical Computer Science*, 28 July 2009, pp. 15–31.

ABOUT THE AUTHORS

LUIZ EDUARDO G. MARTINS is a professor of software engineering at the Federal University of São Paulo. His research interests include requirements engineering, embedded systems, safety-critical systems, and model-driven software development. Martins received a PhD in electrical engineering from the State University of Campinas. Contact him at legmartins@unifesp.br.

TONY GORSCHEK is a professor of software engineering at the Blekinge Institute of Technology. His research interests include requirements engineering, technology and product management, lean product development, quality assurance, and innovation. Gorschek received a PhD in software engineering from the Blekinge Institute of Technology. He's a member of IEEE and ACM. Contact him at tony.gorschek@bth.se; www.gorschek.com.