Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

THE USE OF SIMULATION TO IMPROVE TRUST AND ADOPTION OF AUTONOMY AND AI IN HIGH-CONSEQUENCE WORK SYSTEMS

Emily Barrett
Kelly J. Neville

Modeling and Analysis Innovation Center
MITRE
7515 Colshire Drive
McLean, VA 22102, USA

Lisa Weinstein Billman

Cyber Effects and Information Warfare
MITRE
4801 NW Loop 410
San Antonio, TX 78229, USA

Theresa Fersch

Modeling and Analysis Innovation Center
MITRE
202 Burlington Road
Bedford, MA 01730, USA

Valerie J. Gawron

Transportation Human Centered Experimentation
MITRE
7515 Colshire Drive
McLean, VA 22102, USA

Emily S. Patterson

School of Health and Rehabilitation Sciences
The Ohio State University
453 W. 10th Street
Columbus, OH 43210 USA

Eric S. Vorm

Naval Air Warfare Center
Aircraft Division
22347 Cedar Point Road
Patuxent River, MD 20670, USA

ABSTRACT

We assert that simulation should be an integral part of technology development and acquisition. Its use to iteratively evaluate new technology across the development timeline can help ensure technologies contribute to resilience in work operations. This, in turn, benefits trust and likelihood of adoption. Potential hindrances to simulation in technology development are the time and complexity simulation can introduce. Time may be needed to model entities and dynamics to be simulated, plan and conduct simulation-based tests and experiments, and translate the results into requirements, user stories, or other inputs to the technology's design and implementation plan. Complexity is increased when simulation results suggest new or changed requirements, identify technology design and implementation improvements, or produce conflicting feedback from potential users. We will discuss these challenges, methods and tools that minimize their disruptive effects, varieties of simulation we have used to support technology development, and benefits of using simulation in development.

1 INTRODUCTION

Upon completion and delivery, many new technologies successfully perform the functions they were intended to perform. It is difficult, however, to anticipate how a new technology, even a technology considered successful by most standards, will behave and interact when part of a socio-technical *work*

system. (We define *work system* as a combination of interacting elements organized to achieve one or more stated purposes within a domain of work, with elements including technology, people, policies, protocols, and procedures [based on NIST SP 800-37 Rev. 2 from ISO/IEC 15288].) This difficulty may be the reason only a subset of successfully developed technologies perform well when functioning as part of a work operation responding to realities of work demands and conditions (Neville et al. 2008; Dwivedi et al. 2015; Greenhalgh et al. 2017; Kheybari et al. 2020). Even fewer technologies will perform well with work operations facing unusual realities, such as unusually high demands, a failed key element of the work operation, or environmental conditions that interfere with system performance. A technology that cannot perform its function in coordination with the larger work system across the range of conditions it faces jeopardizes the work system's mission and the safety of people operating and depending on the system. Yes, current technology development processes do not effectively anticipate how well technologies will integrate into a work system's operations and its responses to challenging demands and situations.

1.1 Inadvertent Effects on Resilience

Just as it is difficult to predict how a new technology will perform in work operations, it is difficult to anticipate how a new technology may change the balance among a work system's competing priorities. When a new technology's effects on work system operations go unexamined during development, trust and adoption likelihood are likely to suffer. They are likely to suffer because of the real risk of unintended harmful consequences on work system operations. In particular, common technology modernization and development goals often come at a cost to *resilience,* the ability to adaptively adjust functioning in response to challenging and nonroutine events and conditions and thereby continue to achieve system goals (Nemeth et al. 2011).

A new technology is often developed to fill a mission-critical gap or need. But when introduced into the work system, it will do more than just fill that particular gap or need. It will interact with other work-system and environment factors, potentially affecting the ways they interact with each other. In so doing, it can advantage certain work system goals, both local and global, to the disadvantage of other work system goals. For example, new technologies and other changes intended to strengthen a particular system capability will tend to reduce the system's ability to respond to a wide array of challenges, i.e., specialization trades off with generalizability. A loss of generalized capability also means less resilience.

System design decisions that affect a system's ability to be resilient tend to have an opposing effect on other common system design priorities (e.g., Hoffman and Woods 2011). In addition to affecting (and being affected by) design decisions that strengthen specific capabilities, a system's resilience potential tends to be reduced by design decisions that improve system efficiency, security, compliance with procedures, and the cost of routine operations, among others. Alderson and Doyle (2010) elaborate: "a system with high efficiency (i.e., using minimal system resources) might be unreliable (i.e., fragile to component failure) or hard to evolve" (p. 840). Changes to improve system security often limit who can access and do what. Such limits can be problematically constraining in contingency situations when time is of the essence and crew members may need to back up or fill in for one another. Similarly, changes made to improve compliance with procedures can reduce maneuverability and impeded resourcefulness when responding to contingency and other challenging situations.

Another common tradeoff within work systems is between centralization and decentralization of authority and, along with that, between distant and local authority over decisions and actions. New technologies can change the balance of this authority tradeoff. For example, automated decision making and decision recommender systems can provide a means for distant and central authorities to reach down into frontline operations to influence or determine frontline decisions. Decisions of distant and central authorities tend to be less specialized and responsive to changing demands and conditions relative to frontline, locally made decisions. A shift toward distant and central authority is thus a shift toward less potential for resilience to time-sensitive challenges.

Shifts in the balance among competing work system priorities, such as shifts between resilience and efficiency, security, compliance, and control, should be made with intention. Yet, they tend to be

unanticipated and inadvertent, with their effects unexamined. Their consistency with the values and culture of the work system is likewise typically unexamined. Alderson and Doyle (2010) argue that technologies are "inadequate and incomplete if the human actors do not have their incentives aligned with the technology" (p. 251). We argue that technologies are much more likely to be trusted and adopted when deliberate steps are taken to evaluate and ensure their alignment with or neutral effect on work system goal priorities and values.

1.2 Using Simulation to Minimize Inadvertent Effects

We assert that difficulty in predicting and understanding how a new technology will interact with and affect work system operations is a significant contributor to low levels of trust and adoption by work system personnel. New technologies can pose a threat to work systems when their functions have not been adapted to the realities of work operations or sources of work system resilience to those realities, as well as when they interfere with the balance among competing system goals. This is particularly true for high-consequence work systems for which resilience contributes to avoiding and minimizing mission failures and mishaps with grave and far-reaching effects. Personnel in high-consequence work systems especially may be unlikely to trust or adopt technologies when they have not seen evidence of their ability to work cooperatively with the work system as it faces operational realities and challenges.

The premise of this paper and panel is that iterative use of simulation across a technology's development provides a means for adapting a new technology to the realities of high-consequence work operations and being deliberate about the trade-offs made between resilience and other work system goals. In the section that follows, we provide an overview of popular systems development frameworks and highlight a persistent blind spot related to the work-system integration shortfalls described above. We then discuss the short-lived Department of Defense (DoD) initiative called *Simulation-Based Acquisition* and hypothesize about reasons for its abbreviated lifespan. Panelists' position summaries follow.

1.3 Themes in Technology Development Practice

The Waterfall and 'V' Models of the systems development life cycle established early requirements specification as a gold standard (Balaji and Murugaiyan 2012). Due largely to the realities of changing requirements and an increasing appreciation for the unavoidability of change, new models were introduced. In particular, the Spiral Development Model proposed an iterative process by which a new technology was iteratively assessed and increasingly refined (Boehm 1988). However, the details of the Spiral model tended to be misunderstood or ignored by most, and it was often mis-executed as "a sequence of incremental waterfall developments" (Boehm and Hansen 2000, p. 8).

For a new technology to function in coordination with work system operations, it is essential that across the development life cycle, the technology development team continue to elicit information about the technology's context and interactions and manage the resulting requirements changes. The ways a technology's design is and is not responsive to the realities of work operations are often not revealed until work system personnel have the opportunity to interact with prototypes and consider their use during operations (Deal and Hoffman 2010). When specified at the front end of development, requirements will not be informed by those key aspects of work operations. Frontend requirements also frequently do not benefit from an understanding of how the work system responds to nonroutine demands and conditions. Consequently, a new technology may be verified as meeting its requirements without knowing how it will respond or interact during nonroutine demands and conditions (Woods 2017). Technologies based on requirements identified during frontend analyses are thus likely to be desynchronized and incompatible with the realities of work operations at the time of delivery.

The Agile development method and Development-and-Operations (shortened to DevOps) approach to agile development are relatively recent attempts to move away from upfront-requirements specification as well as cumbersome documentation practices that make requirements difficult to change. Both approaches advocate lightweight, agile communications and documentation, iterative evaluation, stakeholder feedback,

and requirements that are adaptive to changing conditions and feedback (Mohammad 2017; Williams and Cockburn 2003). Although the Agile and DevOps models reduce the likelihood of producing a technology with invalid requirements, we assert that they carry forward a major blind spot of their predecessors. Specifically, despite improvements over predecessor models, we find that as practiced, iterative evaluations and feedback tend not to be about the technology's integration into the work system and its operations. We find that collecting feedback about how a technology integrates into the complex sociotechnical dynamics of a work system, particularly as the work system responds to high-demand and contingency conditions, is generally not practiced during development. Feedback is instead often focused on software "bugs" and the periodic internal assessment of features to determine which the next iteration's prototype should include. And feedback tends to focus, rather than on sociotechnical work system operations, on the technical parts of a work system: decomposed functions, tasks, features, and associated data and information flows. This view is important but also inadequate for understanding how a new technology might interact with "work as done", i.e., with the realities of operations under a variety of both routine and nonroutine conditions. Notably, The *Practitioners' Cycles Model* (Deal and Hoffman, 2010) formalizes a model of iterative and agile development that involves close collaboration with work operations personnel and other stakeholders. Unfortunately, the model remains largely unrecognized and unused.

We assert that overcoming this 'work-as-done' blind spot in technology development methodology and life cycle models is fundamental to achieving trust and adoption of new technologies in work operations. We further argue that simulation methods offer a means for doing so.

1.4 What Happened to Simulation-Based Acquisition?

Simulation-based acquisition (SBA) was an initiative launched by the U.S. Department of Defense (DoD) around 1996. The DoD defined SBA as "an acquisition process in which DoD and industry are enabled by robust, collaborative use of simulation technology that is integrated across acquisition phases and programs" (Modeling and Simulation Acquisition Council 2000). The goal was to leverage simulation capabilities and tools to enable evaluation and maturation of new design concepts. SBA also allowed all acquisition life cycle activities—maintenance, training, test and evaluation, and others—to be proactively evaluated so they could inform the technology's design and implementation.

In other words, SBA enabled a new system or technology's design to be aligned with the larger system of systems surrounding its operation. It included the use of constructive, virtual, human-in-the-loop, and wargame simulations across the technology development life cycle (Zittel 2001). It supported the engagement of stakeholders, including future users (Faye et al. 2001) by allowing them to observe and interact with an evolving version of a new technology in an operational context. It meant the technology's development benefited from exposure to work operations and operator feedback. As noted by Johnson et al. (1998), "The warfighter only has a general notion of his requirements up front and is unable to give a detailed description early on. M&S [*i.e., SBA*] enables the user and the developer to walk up the spiral development ladder together" (p. 4-2), and "virtual systems can participate in exercises and experiments to determine their effect on the battlefield" (p .4-5). Zittel notes that in the SBA of the Joint Strike Fighter, simulation was used to, among other things, negotiate trade-offs among five military services and a number of competing demands for capability.

We suspect that technology development teams and the acquisition community in general were not prepared at the time for the complexity that simulation introduced into the development process. Zittel (2001) describes ways established acquisition process and culture would need to change for SBA to successfully to take hold. This included the need for "an evolved acquisition culture" (p. 126), iterative design, consideration of multiple design options, and the need to cope with continual change and evolution, together with collaboration and seamless integration of the engineering disciplines. These changes may have been too much to ask of an acquisition community largely still executing according to the waterfall model. For example, in his description of the Joint Strike Fighter SBA program, Zittel refers to the challenge of keeping waterfall tendencies at bay: "Just keeping all the traditionalists from finalizing the requirements was a major feat, and some claimed a counter-productive one" (p. 128).

2 PANELIST STATEMENTS: USING SIMULATION TO GAIN TRUST AND ADOPTION

Panelist statements below describe ways to use simulation to benefit technology development that ultimately contribute to operator and stakeholder trust and adoption. In particular, they describe methods for using simulation to iteratively adapt technologies to realities and demands of high-consequence work operations. The methods are intended to be lightweight and complementary to agile and iterative development processes. They include:

- tabletop exercises (TTXs) that produce user stories,
- modeling and simulation to evaluate how design choices interact with work system sociotechnical dynamics,
- gamified work domain analysis, and
- automated usability tests of agile prototype iterations.
-

Method overviews are followed by a review of lessons learned regarding the use of simulation to evaluate technology designs in socio-technical work contexts.

2.1 Generating User Stories Through the Use of Table-Top Exercises by Theresa Fersch

TTXs have existed for centuries in military planning contexts. More recently, they have been leveraged to explore dynamic problem sets and challenges across new domains such as space, cyber, transportation, and healthcare. Exponential increases in the pace of technological change, disruption, and adversary capability highlight the need for a light-weight simulation methodology to enable continual refinement of assumptions, methods, and designs. We also need a means to leverage different perspectives to pro-actively identify and analyze design and implementation challenges and opportunities while simultaneously maintaining a relatively low-cost and scalable approach that can be adapted to available resources.

2.1.1 IDEAS Methodology

Shortly after the terrorist attacks on September 11, 2001, our team was responsible for running complex tabletop exercises designed to improve investigative and analytical techniques. As we matured and refined these exercises over the course of 14 years, a more formal exercise design process began to take shape. Our team established a formal TTX program known as Intelligence Driven, Exercises and Solutions (IDEAS©). We have since applied IDEAS© to other challenges across new domains. The method became more tailorable and scalable to address the dynamic needs of those who would benefit from TTXs. Today, we develop and conduct IDEAS TTXs and other forms of light-weight simulation in nearly all matters pertaining to national interests – cyber, healthcare, transportation, defense, intelligence, infrastructure, emergency preparedness and response, etc.

2.1.2 IDEAS Methodology Adapted for Agile Engineering

In recognition of the value TTXs offer to the iterative adaptation of technology to a work system and its operating conditions, we have produced a version of our TTX methodology specific for supporting AI development and acquisition. This is in response to the hypothesis that a failure to consider the resilience of technology and the environments in which they will be used, during the development of that technology, can result in failed technology transition (Neville et al. 2022). Our new TTX methodology is a linchpin of the Resilience Aware Development (RAD) paradigm developed by the IDEAS team and our colleagues.

The idea behind RAD is that a technology's effects on work system resilience must be considered throughout its development, especially in cases of high-consequence work. Our adapted TTX approach, the RAD Exercise (RAD-X) methodology, was developed to be a lightweight and rapid tool to meet the needs of agile and iterative development approaches (Dorton et al. 2020). The overarching goal of the RAD-X is

to enable technology developers to proactively explore how their technology might affect work system resilience, and to directly translate TTX outcomes into the development lifecycle. More specifically, the goal of RAD-X is to generate user stories (i.e., software requirements) that would support work system resilience when implemented.

The basis for RAD-X is the five resilience factors outlined in the Transform with Resilience during Upgrades to Sociotechnical Systems (TRUSTS) Framework (Neville et al. 2021; Neville et al. 2022). The TRUSTS Framework is an evidence-based synthesis of factors and subfactors that contribute to the resilience of work systems. Resilience here, is defined as a system's ability to self-protect, self-organize, adapt, and evolve its behavior in response to challenges and demands. The overarching goal of the TRUSTS Framework is to enable RAD by specifying for developers the sources of work system resilience that their technology's design or implementation may interact with and impact.

2.1.3 RAD-X Methodology

By using the 3-part RAD-X methodology, technology development teams can iteratively simulate edge cases, i.e., challenging events or conditions, and assess their technology's design and implementation plan across demanding operational contexts throughout an agile or iterative development process. This exploration enables teams to elicit new resilience-based user stories which will inform future development and implementation. The RAD-X Protocol (RAD-XP) is a supplemental guide describing how to conduct each phase of the RAD-X and also providing examples of what types of scenarios and injects will help explore resilience factors. The RAD-XP protocol is outlined in Figure 1.

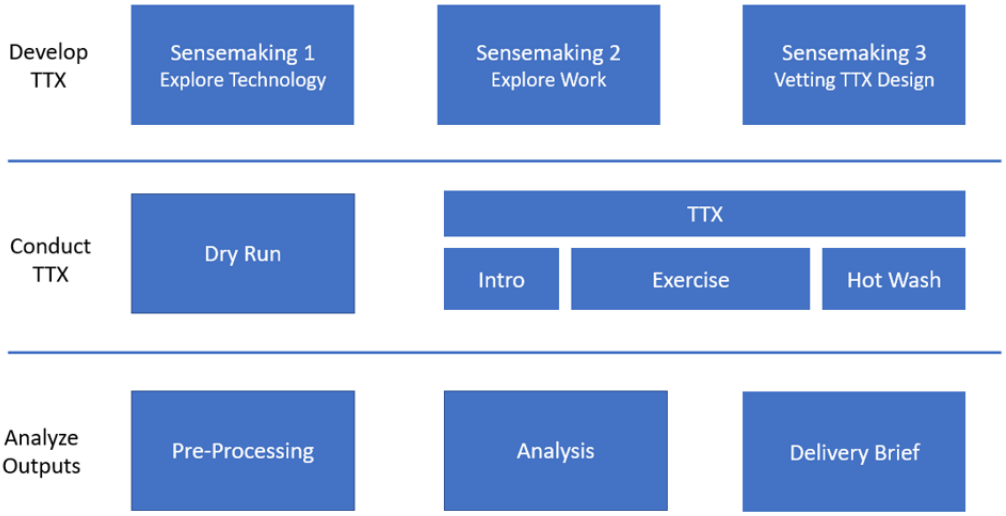

Figure 1: The three iterated phases of the RAD-X methodology.

Phase 1: Development TTX. The first phase of using RAD-Xs in conjunction with a development effort involves three consecutive sensemaking sessions. In these sessions, the RAD-X lead and team aim to understand the purpose of the technology and identify example challenges to use for assessing the technology's role in work system resilience. The exercise lead could be a member of the development team or an external domain expert.

A short but realistic scenario pertaining to each edge case is developed along with 5-6 injects. Injects are major events within that scenario which will drive discussion during the exercise. Injects involve the loss of one or more of the sources of system resilience represented in the TRUSTS Framework. These exercises are designed to be fast paced, with targeted discussion questions to ensure they last no longer than 90 minutes in total. The scenario, injects, and discussion questions are designed with the desired outcomes

of the exercise in mind—user stories that can be addressed in development to improve a technology's ability to participate in work operations during challenging events and conditions.

The first two Sensemaking sessions may not be required or as in-depth in future iterations once the exercise lead has a solid understanding of the technology and the environment in which it functions.

Phase 2: Conduct TTX. The second phase of the RAD-X method involves conducting the exercise. Exercise players should include both technology developers and users. The role of the former group is to represent the performance of the technology; the latter group is best able to recognize resilience related gaps in the design. The exercise lead, usually the lead during the exercise development phase, will serve as the facilitator of the event. There should also be 1-2 note takers. Because many development teams are distributed, these exercises are designed to be conducted virtually. It is advisable to record the events to ease the data analysis process which comes next.

Following an orientation to the new technology to be discussed, the lead describes the scenario, which consists of the day's work conditions, constraints, and goals, highlighting any differences from routine conditions and goals. The lead then introduces the series of injects, or challenges. Each inject's description is followed by discussion questions asking participants to consider how the technology might interact with a work unit's response and ability to persist in achieving their goal(s).

Phase 3: Analyze Outputs. Following each exercise, the final RAD-X phase consists of analyzing the exercise outputs. This includes reviewing the notes taken and recordings of the event. The RAD-XP provides step-by-step instructions for translating RAD-X data into user stories. We have demonstrated the ability to accomplish this translation by two team members in four hours following a 90-minute exercise (Dorton et al. 2023). These user stories are then briefed to the development team and exercise players in the exercise debrief. By mapping user stories to the work-system resilience sources they impact, we contribute to the development team's prioritization of user stories as they determine which to work on in the next development cycle. Understanding improvements to be made to the technology and the impact of those improvements on work system resilience may also be beneficial when justifying resource needs to decision makers.

To summarize, the RAD-X methodology is designed to help development teams evolve their technology designs by exposing it iteratively to versions of challenging work operations. These iterations simultaneously provide an opportunity to consider implementation strategies and evolve the work system to better take advantage of the technology's capabilities. The co-evolution of technology and work system achieved via RAD-X should contribute significantly to trust in the technology's ability to perform across conditions and therefore to adoption. The methodology helps fill a problematic gap in technology development: the lack of methods for considering how a technology will perform as part of a work operation when that work operation is under duress. Importantly, it does so in a way that imposes minimized time and effort requirements on an agile or iterative development team.

2.2 Computational Modeling by Emily Barrett

Over time and with advances in communications and information systems, the high stakes work systems on which we rely, including healthcare, air traffic management, utility, finance, and law enforcement have grown in complexity. As technology continues to play an increasing role in these work systems and offer new and enticing capabilities, the need for dynamic modeling and analysis of work system dynamics becomes imperative. Examples of these dynamics include agent interactions, coordination strategies, task load, etc. There is no perfect model for how these dynamics should look and operate for any system in any environment. For this reason, using computational work models to simulate these work dynamics can help system designers identify key gaps in their system requirements and how best to overcome significant challenges.

As part of MITRE's on-going TRUSTS program, the research team has identified five fundamental resilience factors that contribute to a system's resilience during stressful scenarios (Neville et al. 2021; Neville et al. 2022). Development teams do not have unlimited time and resources to commit to perfectly incorporating and executing all five factors. This computational work model can help simulate high stress

scenarios and produce the quantitative data to analyze the tradeoffs of maximizing efficiency versus maximizing system resilience. The data elicited from this model can also assist in prioritizing certain resilience factors when developers do not have the time or resources to invest into incorporating all five factors.

Computational modeling and simulation of work systems opens a new perspective on cognitive work analysis and analyzing the dynamics of work in an ever-changing and evolving environment. (Ijtsma 2019) Analyzing how a system handles a given scenario in simulated time as opposed to real-time offers "what-if" capabilities, allowing system designers the ability to iterate over many different system design choices.

The modeling environment used in this work is Work Models that Compute (WMC). WMC is a scenario-based simulation environment used to simulate and analyze the collective work performed in a multi-agent system. The application of this framework enables the analysis of task allocation and information exchange in work systems during simulated operations. Through the use of model components like resources, actions, and agents, WMC can be used to predict and quantify human performance requirements given different system architectures (Ijtsma et al. 2018).

WMC uses four different computational structures to fully develop the work model for a system: actions that describe work processes, resources that describe the state of the environment, decision actions that use current conditions to select strategies, and functions that allow for "compositional modeling at multiple levels of abstractions (Pritchett et al. 2014). By manipulating these different WMC structures, we can run experiments to analyze how the system performs with and without certain automated features as well as with and without specific resilience factors. The data elicited from these experiments will provide insights into when the system is pushed beyond its boundaries, how well it recovers and adapts, and how the different agents are coordinating to establish resilient and/or efficient operations.

2.3 Injecting Recommendations to Enhance Work System Resilience Through Gamified Knowledge Elicitation by Emily S. Patterson and Eric S. Vorm

Disruptive technologies in complex systems have ripple effects on the roles and responsibilities of practitioners, sources of resilience and expertise, how conflicting goals for different stakeholder groups are negotiated and reconciled, how resources are shared, and how guidance for action (typically policies and procedures) is designed and updated. When there is no existing base of Subject Matter Experts (SMEs) for knowledge elicitation that closely maps to envisioned roles, we can still learn from carefully selected experts with relevant experience. When new technologies are not fully envisioned, requirements have not yet been written, and important decisions have not yet been made as to the hardware and software characteristics of a new technology, we can still take advantage of knowledge of how previous systems have challenged adaptivity, situation awareness, resilience, coordination, and learning about patterns through modifiable design choices.

We propose a gamified approach to rapidly developing the understanding of work domain challenges and dynamics that is fundamental to new technology design decisions. This approach presents SMEs with context and situations to produce a game-like simulation that facilitates elicitation of this essential baseline.

To prepare for a gamified elicitation session, we recommend eliciting from a SME domain-specific and technology-tailored examples of generically described challenges that are intended to stress the work system. This approach will reveal unwarranted assumptions and potential flaws or gaps in technology design. It will also realign intent of system use with how to best resolve tradeoffs among conflicting goals in specific scenarios and elicit critical information to help group and prioritize this information on primary information screens. Specifically, we propose providing a short description of an envisioned technology as well as the level of embedded automation in the technology. Following this description, we recommend the following questions:

1. Do you have any questions about the technology?
2. Do you have any immediate reactions to share about how the technology is envisioned? Does it seem like it would be useful?

Subsequently, the elicitation game would begin by using a random number generator to select a *scenario event* with a description like the example below that probes the topic of false alarms. Either the specific questions to ask would be randomly selected or the person setting up the scenario run would select which question(s) to use during the game. The numbered elements in the questions were identified during a setup session with a domain SME who is working with the team to contextualize the game to a domain.

Topic	**Event**	**Questions**	**SME prep questions**	**Anticipated recommendations**
False alarms	There is an alarm [1] that is correct about 20% of the time. One of the main reasons for false alarms is that the threshold is set to avoid missing any events, at the cost of a high false alarm rate. [2-3]	1. What alarms do you think would be helpful in this role, [4] even if most of the time they are a false alarm? 2. When would it be important enough to capture a possible event with an alarm [1], even if it is rare? Please fully explain your thinking.	1. What is the alarm? 2. What is the threshold setting that has a lot of false alarms? 3. What is a threshold setting that has few false alarms? 4. What is a role that needs to get this alarm?	1. Ability for an individual in a role or a unit to change the threshold setting for all monitored systems or an individual monitored system 2. "Smart" algorithms that optimize threshold settings based on input values for correct vs. false alarms for a parameter and analysis of how many alarms go off at each threshold 3. Tailor thresholds for cohorts of systems or situations based on the differing risk profiles for missing true events

At the conclusion of a session, these debrief questions are asked:

1. From all the things that we have discussed, what is your highest priority for improving the design of the technology?
2. How useful do you think the technology is on a 1-10 scale from not useful at all to extremely useful? Please tell me more about that.
3. How engaging was this session on a 1-10 scale from not engaging to very engaging? Please tell me more about that. What would have made it more fun?
4. How might we train differently for this new technology?
5. If we eliminated [expensive high expertise person] or had a much higher workload for [expensive high expertise person] because of the new technology, what are your thoughts about that?
6. Anything else that you would like to share?

The 60-minute sessions with individual participants are automatically transcribed. For every session, an initial analysis showing the probe and selected transcript text is estimated to take 90 minutes. Analysis teams composed of human factors experts, systems engineers, and SMEs can classify transcripts into themes and generate consensus recommendations for improving the technology. Responses to the initial and debrief questions are anticipated to provide insights into the reasonableness of assumptions for use, anticipated roles, training, credentialing, staffing, anticipated policies and procedures, ideal prioritization

of conflicting goals for particular scenarios, and information required to support troubleshooting brittle systems with embedded automation that has the potential to need increased transparency.

We acknowledge the intellectual contributions of Jennifer McVay, Lawrence Mize, Emilie Roth, Joseph Gerard, Olivia Hernandez, David Worden, Greg Witkop, Misty Blowers, Sean Guarino, Laura Smith-Velazquez, Alex Morison, Marisa Bigelow, Mike Nicoletti, Michael Smith, Grit Denker, and Anil Raj.

2.4 Automating the Assessment of Usability by Lisa Weinstein Billman

The need to develop, test and deploy software rapidly has driven the Air Force to employ agile methodologies. Agile software development is a set of methods and practices where solutions evolve through collaboration between self-organizing, cross-functional teams. Between the time system requirements are defined and the fielding of a system, all sorts of changes in the work operation and its environment may render those requirements invalid. Conducting agile software development allows for those changes to be accounted for in the design at any time throughout the development and fielding of a system by eliciting feedback periodically from stakeholders and end users.

In a user-centered design scenario, user feedback provided early and often in the design and development cycle drives design, rather than only engaging users at the end of development to critique a finished product. The result is a more usable product that better meets customer expectations with minimal redesign at the end. As shown in Figure 2, requirement changes early in the design process are generally easier and less costly to make. The implementation of an agile software development approach along with automating usability testing can allow developers to modify requirements to meet evolving work conditions and repeat tests under the new conditions to ensure the human-system-environment will perform as intended. The automated test results, and design modifications will increase trust of the system when fielded and ensure optimized human system environment performance. This automated approach also allows repetitive testing under different conditions and can be applied to many aspects of usability testing.

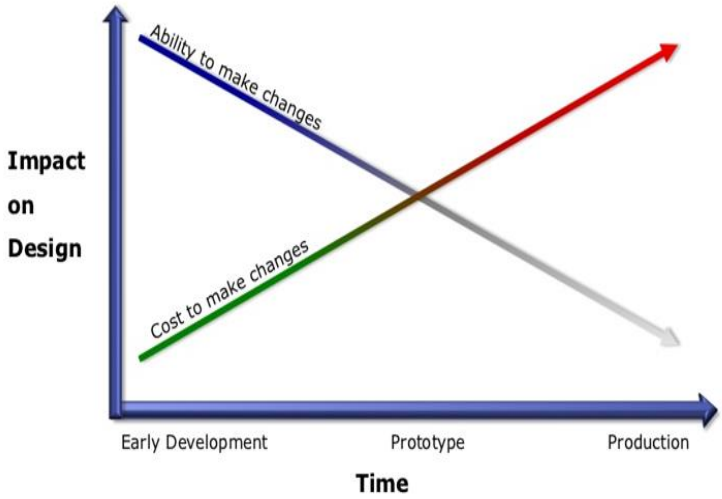

Figure 2: Impact of changes in design over time.

Usability testing assesses whether operators will be able to learn and use a system, with minimal cognitive load and human error. Potential human-system-environment performance issues identified during usability testing are handled as system deficiencies and entered into the backlog to be addressed by the designers and developers. Usability assessments are often conducted by Human Factors Engineers (HFE) eliciting feedback from end users, and/ or by the HFE analyzing the product for typical usability issues. These issues generally arise from violations of basic heuristics or rules of thumb for good usability (Nielsen, 1994).

Usability considers the technology that is being developed as well as the human user that will interact with the technology. Suitability adds an additional consideration to the development of products by ensuring the environment in which the human operator will use the technology is considered in the development process. For example, will the system be used in bright sunlight or with an operator wearing gloves? If so, a glare resistant screen may be needed and a touch screen may not be optimal for input. As shown in the Venn diagram below, assessing all three components ensures the system has the highest suitability.

A critical aspect of agile software development is to automate testing, so that small pieces of code can be tested quickly, frequently and without human intervention. If portions of code are developed and tested daily, it is necessary to ensure that all aspects of testing are completed before the new code is added to the baseline. If the new code involves any user interface development or the workflow for the operator, then those changes need to be tested for usability and suitability. Figure 3 shows that usability and suitability testing must evaluate the interactions of users with technologies and systems within the operational work environment.

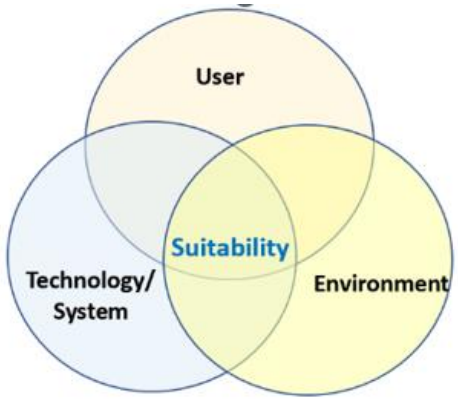

Figure 3: Intersection that is the focus of usability and suitability testing.

Due to the time required to conduct human in the loop usability/suitability testing, it is not feasible to conduct these tests as part of the continuous testing process employed with agile development. Therefore, it is desirable to find a method for automating some parts of these evaluations so that these tests are done continuously along with functional and integration testing. Automating usability testing will allow many usability issues to be found early and corrected through the regular processes in an agile development environment.

Although some usability purists may argue that usability testing cannot be performed without a human in the loop, there are numerous usability requirements that can be tested using automation. One simple example is the number of actions (keystrokes) required to perform a function. If a threshold of keystrokes is set for an emergency action, then it is simple to test compliance with that requirement automatically. Another would be to measure consistency of font type, font size or color usage. MITRE has developed a prototype automated test tool that can assess compliance with requirements such as these that impact usability. Similar test scripts could be developed to assess compliance with suitability requirements such as contrast between text and background to accommodate environmental lighting, or size of touch screen buttons to accommodate gloved users. Ultimately, the goal is to be able to automate a larger portion of usability and suitability testing and improve the final software product.

The addition of automated test scripts to development pipelines is the first phase in the evolution of automating usability and suitability testing. More advanced concepts would involve the creation of a human agent (artificial intelligence) that imitates a human interacting with software systems and measures the agent's responses. Negative responses would trigger an alert to a usability designer to assess and improve the software. As we continue to research how the human brain reacts to well-designed or poorly-designed

systems, we can start developing mental models of code to emulate user performance. Once we can determine how to emulate the human interacting with the software, we can begin to collect automated usability test data. The end user will always play a critical role in the assessment of software systems, but the use of automated testing can reduce the amount of time required by users and enable them to focus on the most impactful aspects of system design.

2.5 Lessons Learned in the Use of Simulation to Achieve Earned Trust and Adoption of Autonomy and AI in High-Consequence Work Systems by Valerie J. Gawron

George Santayana (1905) stated that "Those who cannot remember the past are condemned to repeat it." The following lessons learned are based on my 43 years of applying and evaluating simulation. Designers of sensor fusion, decision aiding, and autonomous systems have extensively applied Artificial Intelligence (AI) and expected these systems to improve situational awareness (SA) and hence mission effectiveness. This has not always been the case. A study of such cases resulted in eleven lessons learned. Some could have been avoided by the use of simulation during technology development. Others relate to tools and methods for assessment during simulation-based evaluation. Each lesson is described below.

One – There may be a dissociation among SA, workload, and performance. The inverted U relationship between workload and performance has been known since the 1920s. Specifically performance is optimum at moderate levels of workload and degraded at either low or high levels of workload. There seems to be a similar relationship between SA and workload, i.e., optimum SA occurs at moderate workload. At too little workload, a person minimizes his or her sensory processing. At too high workload, a person focuses on only one stimulation. Simulation can be used to vary operator workload, enabling the evaluation of new technology impacts on SA and performance across workload levels (Gawron and Lehman 1992).

Two – Operators can have too much SA. For example, a sensor fusion display presented all relevant information on all threats that could destroy the aircraft carrying the sensor fusion. Pilots' performance measured as the distance penetrated past the threats degraded relative to a system that showed only a small portion of the threats (Gawron and Funke 1993). Surprising emergent effects such as this are possible when new technologies are introduced into complex systems. Simulation is an important tool for evaluating the validity of technical requirements; specifically, embedded hypotheses and assumptions about how a technology's design will affect system performance.

Three – All three aspects of SA must be measured during any system evaluation (Lehman et al. 1991). These aspects are: 1) sense entities, 2) identify entities, and 3) project the actions of the entities. A decision aiding system automatically detected and identified all entities. The human operator was expected to project the future actions of the entities. Operators performed worse at projecting when using the system. The reason: operators felt that they did not have the opportunity to assess the behavior of the entity during the identification. This is yet another case where simulation could have been used during development to ensure a technology's requirements are valid in the context of work-system operations.

Four – There are individual differences. This is nothing new. Any pilot knows who has the best SA in the squadron. Ironically the tester expects the best to show the greatest increase in SA due to a new SA-enhancement system. But the best doesn't have much room for improvement (Gawron at al. 1995).

Five – The right intentions translated into a bad design results in bad SA. An adaptive system built extensively using AI algorithms perfectly compensated for decrements in pilot performance but informed the pilot of actions taken using a low contrast display that washed out in bright sunlight (Gawron 2002; Gawron and Fryer 2003).

Six – Who has responsibility is critical to assessing SA. Persons who do not think they are responsible for being aware of an entity do not try to maintain SA of that entity (Gawron 2019a).

Seven – Rules are made to be broken. For example, technology that fuses sensor images should make identification of entities easier. However, in one system, operators felt the work of manually fusing the information enhanced SA of these entities (Gawron and Priest 2001).

Eight – Words are critical in rating SA. The Situation Awareness Rating Technique (SART) was developed for British pilots. Americans have different connotations to some SART words and phrases, such as the "degree that one's thoughts are brought to bear on the situation" (Gawron et al. 2010).

Nine – Perceived potential for grave consequences, including injury and death, is an SA booster. Firefighters describe the phenomenon of every sense being more intense and all action viewed in slow motion. Pilots in flight have higher SA than pilots in a simulator (Gawron et al. 1989; Gawron at al. 1992; Gawron and Reynolds 1995).

Ten – Building schema is critical to SA for humans and computers. Schema are knowledge structures that represent common patterns encountered in the environment and that help cut through data to the information, reducing data processing requirements (Gawron 1997).

Eleven – SA, trust, and workload all must be considered for operators to have high workloads. Further, operator distrust of systems decreases SA and increases workload (Gawron 2019b).

3 DISCUSSION

It is time for simulation, and specifically simulation of technology interactions with sociotechnical work system operations, to become an integral activity throughout technology development and modernization. Current development models tend to be more iterative and accepting of change and evolution than their predecessors. They are therefore more likely to successfully integrate value-added simulation into their processes. And our premise is that work-system simulation, whether high or low fidelity, small or large scale, needs to be a part of their processes to achieve trust and adoption.

As Zittel (2001) reported, simulation can bring with it significant overhead in the form of simulation infrastructure and exercise planning and execution. It can produce large amounts of feedback that leave a development team reeling. It can add complexity to development when iterative simulation results suggest design changes throughout the development process, many of which may require significant rework, potentially at significant cost.

We suggest simulation overhead and complexity costs can be minimized such that they are outweighed by the benefits. In the panelist statements above, we described strategies, methods, tools, and lessons learned that can mitigate obstacles to SBA. And we presented ways simulation can be used to adapt new technologies to the realities of work system operations, including demanding and contingency operations. We argue that the increased use of simulation will improve the ability of personnel to trust new technology to "be there" for them and their work units across challenging, as well as routine, performance conditions.

ACKNOWLEDGMENTS

We would like to express our appreciation to Dr. Saurabh Mittal for the invitation to participate in the "Complex and Resilient Systems Track" of the Winter Simulation Conference. We also thank our reviewers, Ed Hua, Doug Flournoy, Dr. Kris Rosfjord, Beverly Wood, and Dr. Matthijs Broer, and our sponsor, Dr. Kris Rosfjord, for encouraging the development of the vision and methods described in this paper. This work was funded by MITRE's Independent Research and Development Program. Approved for Public Release; Distribution Unlimited. Public Release Case Number 23-2327.

REFERENCES

Alderson, D.L. and J.C. Doyle. 2010. "Contrasting Views of Complexity and their Implications for Network-Centric Infrastructures". *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 40(4): 839-852.

Balaji, S., and M.S. Murugaiyan. 2012. "Waterfall vs. V-Model vs. Agile: A Comparative Study on SDLC." *International Journal of Information Technology and Business Management* 2(1): 26-30.

Boehm, B. W. 1988. "A Spiral Model of Software Development and Enhancement". *Computer* 21(5): 61-72.

Boehm, B., and W.J. Hansen. 2000. *Spiral Development: Experience, Principles, and Refinements*. Technical Report SEI-2000-SR-008, Carnegie Mellon University Software Engineering Institute.

Deal, S.V., and R.R. Hoffman. 2010. "The Practitioner's Cycles, Part 1: Actual World Problems". *IEEE Intelligent Systems* 25(2): 4-9.

Dorton, S.L., L.R. Maryeski, L. Ogren, I.T. Dykens, and A. Main. 2020." A Wargame-Augmented Knowledge Elicitation Method for the Agile Development of Novel Systems". *Systems 8*(3): 27.

Dwivedi, Y.K., D. Wastell, H.Z. Henriksen, R. De. 2015. "Guest Editorial: Grand Successes and Failures in IT: Private and Public Sectors." *Information System Frontiers* 17:11-14.

Faye, P., E.B. Andrew, and J. Lee. 2001. "Extending Simulation-Based Acquisition (SBA) to the Warfighter with the Air Force Joint Synthetic Battlespace (JSB-AF)." In *Enabling Technology for Simulation Science V* 4367: 298-304.

Gawron, V.J. 2019a. "Lessons Lost: What We Learned About Automation in Aviation Can Be Applied to Advanced Driver Assistance Systems and Autonomous Vehicles". *Ergonomics In Design* September.

Gawron, V.J. 2019b. "Simulation Applications in Training". Presentation at the Flight and Ground Simulation Update, September 23rd, Naval Air Warfare Center, Patuxent River, MD.

Gawron, V.J. 2002. "Lessons Learned in the Design and Operation of UAVs". Invited presentation at John Deere, April 9th.

Gawron, V.J. 1997. "Lessons Learning in Using Situational Awareness to Evaluate Systems". In *Proceedings of the Second Annual Symposium and Exhibition on Situational Awareness in the Tactical Air Environment*, edited by K. Garner. Patuxent River, MD: Naval Air Warfare Center Aircraft Division.

Gawron, V.J., R.E. Bailey, and E. Lehman. 1995. "Lessons Learned in Applying Simulators to Crewstation Evaluation". *The International Journal of Aviation Psychology* 5(2): 277-290.

Gawron, V.J., R.E. Bailey, L.H. Knotts, and G.R. McMillan. 1989. "Comparison of Time Delay During In-Flight and Ground Simulation". In *Proceedings of the 33rd Annual Meeting of the Human Factors Society* 120-123.

Gawron, V.J., and W.D. Fryer. 2003. "Simulation Fidelity: Lost in the Details". *The Flyer* Spring/Summer:2–5.

Gawron, V., and D. Funke. 1993. "Lessons Learned in Large Scale Simulation for Training". In *Proceedings of the 37rd Annual Meeting of the Human Factors Society*.

Gawron, V.J., T. Hughes, J. Hassoun, R. Bailey, and S. Horowitz. 1992. "Comparison of a Head Up Display Evaluation in Ground and Flight Simulation". In *Proceedings of the 36rd Annual Meeting of the Human Factors Society*.

Gawron, V.J., and E.F. Lehman. 1992. "Optimizing the Use of Aircrew-in-the-Loop Simulation During System Design and Evaluation". Presented at the *Military Operations Research Symposium 60th Contingency Operations in a New World Order*, Alexandria, Virginia.

Gawron, V.J., G.R. McMillan, and R.E. Bailey. 2010. "The Effects of Time Delay and Physical Motion on Manual Flight Control: An In-flight and Ground-based Simulation Experiment". *The International Journal of Aviation Psychology* 20(3): 221–248.

Gawron, V.J., and J. Priest. 2001. "Night Vision Goggles: Lessons Learned". Poster presented at the 45th Annual Meeting of the Human Factors and Ergonomics Society.

Gawron, V.J., and P.A. Reynolds. 1995. "When In-Flight Simulation is Necessary". *Journal of Aircraft* 32(2):411-415.

Greenhalgh T, J. Wherton, C. Papoutsi, J. Lynch, G. Hughes, C. A'Court C, et al. 2017. "Beyond Adoption: a New Framework for Theorizing and Evaluating Nonadoption, Abandonment, and Challenges to the Scale-Up, Spread, and Sustainability of Health and Care Technologies". *Journal of Medical Internet Research* 19(11):e367.

Ijtsma, M. 2019. "Computational Simulation of Adaptation of Work Strategies in Human-Robot Teams". Dissertation. Atlanta, GA: Georgia Institute of Technology.

Ijtsma, M., L. Ma, K.M. Feigh, and A.R. Pritchett. 2018. "Demonstration of the "Work Models that Compute" Simulation Framework for Objective Function Allocation". In *Proceedings of the Human Factors and Ergonomics Society* 1:321-324.

Johnson, M. V., M.F. McKeon, and T. Szanto. 1998. "Simulation Based Acquisition a New Approach." Fort Belvior, VA: Defense Systems Management College Press.

Kheybari, S., F.M., Rezaie, S.A. Naji, M. Javdanmehr, and J. Rezaei. 2020. "Evaluation of Factors Contributing to the Failure of Information Systems in Public Universities: The Case of Iran". *Information Systems 92*:101534.

Lehman, E.F., V.J. Gawron, W.J. Cody, D.R. Nelson, and M. Jenkins. 1991. "Handbook for Crewstation Simulation Test Program Planning". Technical Report No. HSD-TR-91-0001. Brooks AFB, TX.

Modeling & Simulation Acquisition Council. 2000. *Simulation Based Acquisition Definition.* Washington, DC: DoD.

Mohammad, S. M. 2017. "DevOps Automation and Agile Methodology". *International Journal of Creative Research Thoughts* 5(3):2320-2882.

Nielsen, J. 1994. *Usability Engineering*. San Francisco, CA: Morgan Kaufmann.

Neville, K.J., B. Pires, P. Madhavan, M. Booth, K. Rosfjord, and E.P. Patterson. 2022. "The TRUSTS Work System Resilience Framework: A Foundation for Resilience-Aware Development and Transition". In *Proceedings of the Human Factors and Ergonomics Society 2022 Annual Meeting*. Atlanta, GA.

Neville, K., R.R., Hoffman, C. Linde, W.C. Elm, and J. Fowlkes. 2008. "The Procurement Woes Revisited". *IEEE Intelligent Systems January/February*:72-75.

Neville, K.J., H. Rosso, and B. Pires. 2021. "A Systems-Resilience Approach to Technology Transition in High- Consequence Work Systems". *Paper presented at the 9th Symposium on Resilience Engineering*, June 21-24, Toulouse, France.

Pritchett, A. R., K.M. Feigh, and D.D. Woods. 2010. "Facets of Complexity in Situated Work". In *Macrocognition Metrics and Scenarios*, edited by E. S. Patterson and J. E. Miller, 221-251. Burlington, VT: Ashgate.

Santayana, G. 1905. *The Life of Reason: The Phases of Human Progress.* New York, NY: Charles Scribner's Sons.
Williams L., and A. Cockburn 2003. "It's About Feedback and Change." *IEEE Computer* 36(6):39-43.
Woods, D.D. 2017. "STELLA: Report from the SNAFUcatchers Workshop on Coping With Complexity". Technical Report. Columbus, OH: The Ohio State University.
Zittel, R. C. 2001. "The Reality of Simulation-Based Acquisition--And an Example of US Military Implementation". *Acquisition Review Quarterly* Summer:121-132.

AUTHOR BIOGRAPHIES

EMILY BARRETT is a simulation engineer at MITRE. Her research focuses on resilience engineering and human computer interaction. For her recently awareded Early Career Research Project grant, Emily is extending her graduate research in the computational modeling of work system dynamics to study sources of work system resilience and ways new technology can affect resilience in high-stakes work systems. Her email address is ebarrett@mitre.org.

LISA WEINSTEIN BILLMAN is a Project Leader with The MITRE Corporation overseeing the development of a Joint Cyber Command and Control system for the US Department of Defense. Her focus is the application of Human Systems Integration in the acquisition of complex military systems. Her current work includes the integration of automation and artificial intelligence into cyber systems and the automation of usability testing. Her email address is lbillman@mitre.org.

THERESA FERSCH is the creator and lead for Intelligence-Driven Exercises and Solutions (IDEAS) table-top exercise (TTX) capability at the MITRE Corporation. Her focus is on developing and conducting tailored and scalable solutions that allow experts to explore complex challenges in a meaningful way. She supports work across an array of domains including cyber, health, transportation, climate, defense, and intelligence. Her email address is tfersch@mitre.org.

VALERIE J. GAWRON is a human systems integration engineer at THE MITRE Corporation. Dr. Gawron provides technical leadership in Research, Development, Test, and Evaluation of small prototype systems through large mass produced systems with an emphasis on aviation safety and human performance optimization. Gawron has written over 445 publications including the Human Performance, Workload, and Situation Awareness Measures Handbook (third edition) and 2001 Hearts: The Jane Gawron Story. Both are being used internationally in graduate classes, the former in human factors and the latter in patient safety. Her email address is vgawron@mitre.org.

KELLY J. NEVILLE is a cognitive systems engineer at MITRE. Her work focuses on the dynamics of complex sociotechnical systems, enabling their resilience to challenges in high-stress and high-uncertainty conditions, and the design and evaluation of advanced technologies from a systems theory and resilience perspective. Her email address is kneville@mitre.org.

EMILY S. PATTERSON is a Professor in the School of Health and Rehabilitation Sciences, College of Medicine, at The Ohio State University. She is an associate editor for the Human Factors in Healthcare journal. Her email address is patterson.150@osu.edu. Her papers are at: https://www.researchgate.net/profile/Emily-Patterson-15.

ES VORM is a cognitive systems engineer in the United States Navy. Lieutenant Commander Vorm's research marries traditional human computer interaction and human factors with systems engineering to build robust solutions for high-risk sociotechnical systems. ES Vorm's research advocates for the unique and powerful capabilities of human beings, and seeks to design systems that respect and value those capabilities. He currently serves as the Design of Experiments Lead for the DARPA Enhanced Design for Graceful Extensibility (EDGE) program, capability portfolio manager for the Autonomy Community of Interest at the Office of the Secretary of Defense, and US delegate member of the Human Factors Specialist Team to NATO. His email is esv@esvorm.com and his website is https://www.esvorm.com.