Human-Centered Systems Engineering:

A Unified Approach to Product Safety Engineering

GM Samaras
Samaras & Associates, Inc.
Pueblo, Colorado, USA
george@samaras.eng.pro

Abstract—**This workshop discusses the application of human-centered systems engineering as a defense against human errors resulting in unsafe systems.**

Human-centered systems engineering, human factors, human error, safety engineering, quality engineering

I. Introduction

This 3-hour workshop is not about hardware safety engineering (EMI suppression, preventing electrocution or fire, etc.) or software safety engineering (preventing data corruption, denial of service, etc.). It is about minimizing the occurrence of unsafe human acts – a defense against human errors. It is a structured, systematic engineering approach to identify and minimize use and user errors. It is about the extension of classical systems engineering to human-centered systems engineering.

You can expect to learn in this workshop what is human-centered systems engineering, the interaction of human and system errors, what we mean by human-centered system complexity, and how we can define human-centered system quality, so that all other quality definitions are subsumed. We will analyze the system development and deployment lifecycle in detail and identify how human factors engineering can contribute to all aspects of development and deployment design.

II. Human-Centered Systems Engineering

A. *Human-Centered Systems Engineering*

The term *systems engineering* (SE) was used at least as early as the 1940s at the Bell Telephone Laboratories and possibly earlier [2]. SE is a very powerful mechanism for reducing business and technical risk. SE is a structured, systematic approach to the development, deployment, and replacement of products, processes, and services. These tools (products, processes, and services) are developed and maintained solely because their use by humans has real (utilitarian) or perceived (esthetic) value. Even completely automated, unsupervised tools have human users (maintenance personnel).

Human-centered systems engineering (HCSE) extends SE to emphasize the criticality of *human actors* and their organizations in development, deployment, and maintenance or replacement of tools. *Actor* is a term of art in the social sciences and economics; it subsumes *user* and *customer*. These actors and their organizations are the stakeholders – ALL of the stakeholders, not just customers, users, or "critical" stakeholders. HCSE is an engineering paradigm and is characterized by a state space (Figure 1) and a lifecycle (Figure 2) [11]. The paradigm applies to both development and deployment, both of which involve design activities.

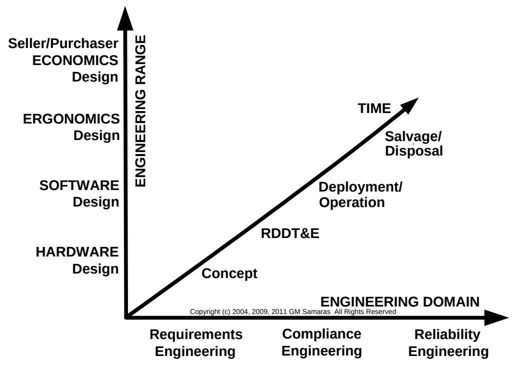

Figure 1: HCSE State Space

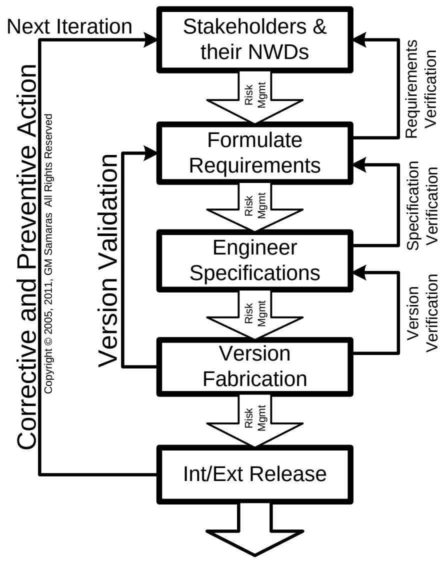
Next Iteration
Stakeholders & their NWDs
Formulate Requirements
Engineer Specifications
Version Fabrication
Int/Ext Release
Risk Mgmt
Requirements Verification
Specification Verification
Version Verification
Version Validation
Corrective and Preventive Action
Copyright © 2005, 2011, GM Samaras All Rights Reserved

Figure 2: HCSE Lifecycle (ISO 14971 §6.3 verifications not shown)

Identifying all the stakeholders, and determining their Needs, Wants, and Desires (NWDs), is a central focus of HCSE (Figure 2). NWDs may be discriminated using Kano's [4] stakeholder response matrix (Figure 3). Missing or misidentifying stakeholders invariably will result in their NWDs being misjudged or overlooked in the design. This results in conflicts among the different stakeholders' NWDs going undetected and often results in stakeholder dissonance [9,10,13]. Stakeholder dissonance (SD) during development leads to development of the wrong tool (stakeholder NWDs are the basis for development requirements). SD during deployment leads to wrong choices of tools and their implementation within the organization, leading to errors, workarounds, and threats to safety and profitability.

	POORLY MET	MET	VERY WELL MET
NEEDS (Basic Needs)	DISGUSTED	UNHAPPY	NEUTRAL
WANTS (Performance Needs)	UNHAPPY	NEUTRAL	HAPPY
DESIRES (Latent Needs)	NEUTRAL	HAPPY	DELIGHTED

Figure 3: Discriminating Stakeholder Needs, Wants, and Desires

B. *Human versus System Errors*

Human errors may be the last bastion of equipment and process safety and effectiveness problems [7]. We have all heard of incidents, near misses, and accidents that are attributed to human error. But we never really hear exactly which human's error is the root cause. The source of human errors can be the result of design & development and/or the result of problems with deployment (user condition, user tool selection, user training, work structure, and the work environment). While it was historically thought that design problems could be alleviated with labeling and training, this is now generally recognized not to be the case.

We generally only hear about four types of human error: Use, unexpected Use, misUse, and abUse. But this ignores the difference between the two separate categories of human error: *user* error that is attributable to the internal and/or external user environment, excluding the tool itself (locus of control: the individual), and *use* error that is attributable to development and/or deployment design (locus of control: development and/or deployment organizations). So, we have human error whose root cause is the human user and we have human error whose root cause is the human developer or deployer of the tool being used.

	ERROR CATEGORY	
ERROR TYPE	USE ERROR	USER ERROR
EXPECTED	ACTIVE (KNOWN BUGS)	ROUTINE USE
UNEXPECTED	LATENT	NEW USE
MISGUIDED	DRIFT	MISUSE
MALICIOUS	SABOTAGE	ABUSE

Figure 4: HCSE Human Error Taxonomy

Figure 4 shows a taxonomy of such errors organized by error category versus error type. In the category of *use errors*, Reason [6] has distinguished *active* errors – the result of known development/deployment "bugs" and *latent* errors – the result of unknown development/deployment "bugs". Dekker [3] identifies *drift* errors – a misguided, typically slow, incremental progression of systems operations taking the tool beyond its originally designed safety envelope. Finally, we have malicious corruption of the tool (*sabotage*). The corresponding *user errors* are *routine use*, *new use*, *misuse*, and *abuse*. Human errors in the use and user categories are not mutually exclusive; very often failures occur when multiple contributors err – each necessary, but only jointly sufficient [15] – resulting in failure. All these human errors must be considered in risk management and safety engineering.

C. *Human-Centered System Complexity*

Introducing human actors into any endeavor dramatically increases the possible number of incorrect or inappropriate responses of a hardware/software system. Human actors, and their organizations, drastically increase system complexity. Complex systems have *emergent* properties that are the result of component interactions at the *interfaces* and that are not readily predictable without appreciation of the system as a whole. Not fully appreciating human-centered system complexity has been an important obstacle in the design and deployment of essential systems. We now recognize that development-induced and deployment-induced errors are a serious problem, may become critical safety issues, and are an important source of reduced quality. They can rarely be mitigated merely with labeling or user training!

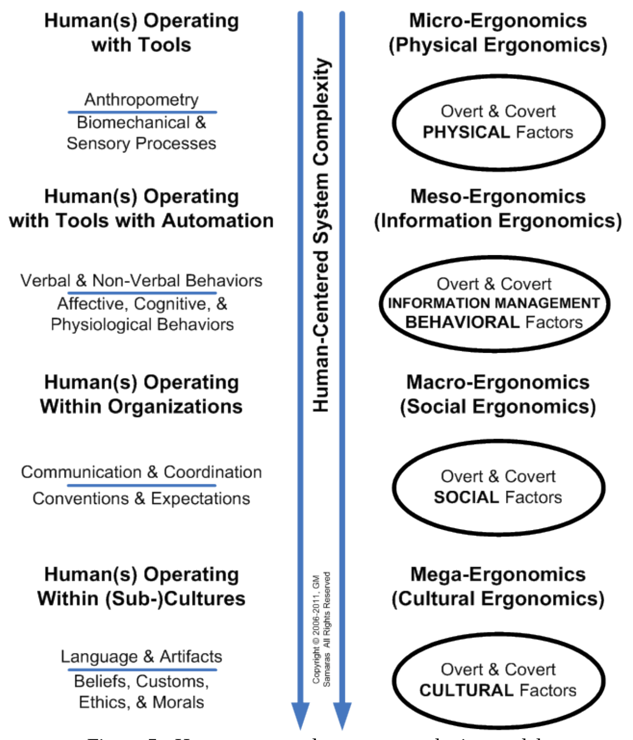
Human(s) Operating with Tools
Anthropometry
Biomechanical & Sensory Processes
Human(s) Operating with Tools with Automation
Verbal & Non-Verbal Behaviors
Affective, Cognitive, & Physiological Behaviors
Human(s) Operating Within Organizations
Communication & Coordination
Conventions & Expectations
Human(s) Operating Within (Sub-)Cultures
Language & Artifacts
Beliefs, Customs, Ethics, & Morals
Human-Centered System Complexity
Copyright © 2006,2011, GM Samaras All Rights Reserved
Micro-Ergonomics (Physical Ergonomics)
Overt & Covert PHYSICAL Factors
Meso-Ergonomics (Information Ergonomics)
Overt & Covert INFORMATION MANAGEMENT BEHAVIORAL Factors
Macro-Ergonomics (Social Ergonomics)
Overt & Covert SOCIAL Factors
Mega-Ergonomics (Cultural Ergonomics)
Overt & Covert CULTURAL Factors

Figure 5: Human-centered system complexity model

A key characteristic of the human-centered approach is that we must achieve a detailed appreciation of the interfaces with and between human actors. Without this, we remain unable to predict and control the critical human and organizational influences on development and deployment, as well as the sensitivities to external factors – which may lead to human errors and safety issues. Our fundamental need to study the system as a whole leads to the model of human-centered system complexity shown in Figure 5 [12]. The four levels of the model categorize the human-system interfaces with and between actors. The first two levels (micro- & meso-ergonomics) deal with interface attributes of individual actors and their tools. The last two levels (macro- & mega-ergonomics) deal with interface attributes between actors operating within their organizations and with their tools. This model recognizes that there are both overt and covert interface attributes and offers a structured, systematic approach to analyzing all the human-system interfaces.

D. Human-Centered Quality

With the emphasis on the identification and discovery of all the stakeholders and their NWDs, HCSE takes a different approach to quality definition and quality management (including quality improvement). HCSE defines quality as *the degree to which the needs, wants, and desires of all the stakeholders have been satisficed* [8]. The term *satisfice* – presumed to be a contraction of satisfy and suffice – was coined by Simon [14] during an attempt to reduce the computational complexity of a linear programming problem for individual and organizational behaviors. *Satisfice* is defined as obtaining a good result, which is good enough but not necessarily the best, for each of all the stakeholders. Satisficing stakeholders is about reducing SD. Therefore, quality (Q) and SD are related concepts; zero SD corresponds to total quality (Q = 1 - SD). Under this formulation [10], total quality (SD = 0) is unachievable, except in the most trivial cases. Quality Management becomes the process of identifying all the stakeholders, measuring their NWDs, and controlling their conflicts; Quality Improvement becomes the process of identifying and reducing SD among all the stakeholders.

From a Quality Engineering perspective [10], managing SD is consistent with: Lean (optimizing value flows); Six Sigma (reducing variability along the value stream); Balanced Scorecard (linking specific processes to organizational strategy); and Quality Function Deployment (translating overall "quality into component quality, individual parts quality, and process elements and their relationships" [1]).
Attempting to meet some or all the stakeholders' NWDs has always been the sole purpose for development and deployment of any product, process, or service. In the transition from SE to HCSE, the emphasis shifts to iterative discovery of stakeholders, identification of their evolving NWDs, and attempsts to reconcile their conflicts, so as to satisfice all stakeholders. This shift in emphasis tends to mitigate errors and omissions early in the development and deployment lifecycles, reducing their final cost. Absent robust HCSE, essential systems will continue to hinder rather than help, be unsafe, ineffective, and economically inefficient; they will continue to be examples of poor quality.

III. HCSE DEVELOPMENT & DEPLOYMENT LIFECYCLE

Both the development process and the deployment process (including maintenance and replacement) influence safety. Both processes can be described with the lifecycle diagrammed in Figure 2. But that does not clearly establish the linkages between these safety-critical processes. Figure 6 is another view of these linked lifecycles [10]. On the far left is the traditional technology development design cycle. It is linked to the deployment design cycle, which in turn is linked to the post-deployment surveillance cycle. Propensity for increasing the probability of human error may occur in each of the three linked cycles and will propagate across the linked cycles. The engineering lifecycle in Figure 2 applies to hardware, software, human factors, and economic engineering (the latter two being subdisciplines of industrial engineering). In this workshop, we will focus on the human factors engineering component, which will drive engineering aspects of hardware, software, and seller/purchaser economics.

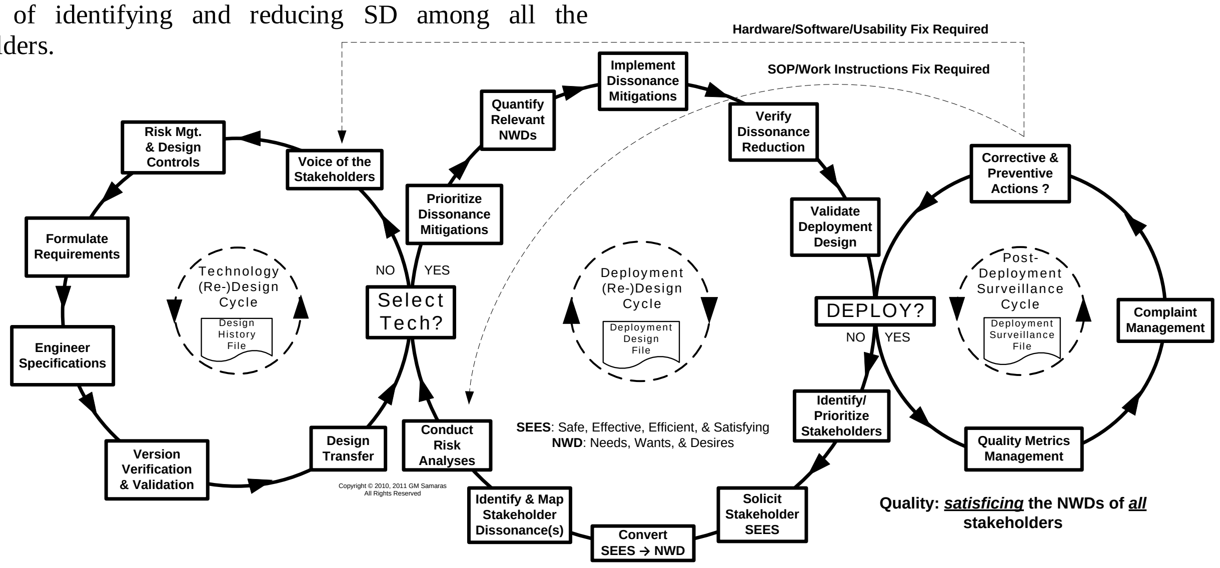
Hardware/Software/Usability Fix Required
SOP/Work Instructions Fix Required
Implement Dissonance Mitigations
Quantify Relevant NWDs
Verify Dissonance Reduction
Risk Mgt. & Design Controls
Voice of the Stakeholders
Prioritize Dissonance Mitigations
Corrective & Preventive Actions ?
Validate Deployment Design
Formulate Requirements
Technology (Re-)Design Cycle
Design History File
NO
YES
Select Tech?
Deployment (Re-)Design Cycle
Deployment Design File
DEPLOY?
NO
YES
Post-Deployment Surveillance Cycle
Deployment Surveillance File
Complaint Management
Engineer Specifications
Identify/ Prioritize Stakeholders
Design Transfer
Conduct Risk Analyses
SEES: Safe, Effective, Efficient, & Satisfying
NWD: Needs, Wants, & Desires
Quality Metrics Management
Version Verification & Validation
Copyright © 2010, 2011 GM Samaras All Rights Reserved
Identify & Map Stakeholder Dissonance(s)
Convert SEES → NWD
Solicit Stakeholder SEES
Quality: satisficing the NWDs of all stakeholders

Figure 6: Linked Lifecycles - Development, Deployment, and Surveillance

The domain of engineering activities shown in Figure 1 are detailed in Figure 7; in all cases they result in test design – for verifications, validation, compliance, and reliability determination. From here, we use the modern, unambiguous terminology for "requirements", "specifications; they are "design inputs" and "design outputs", respectively [9].

Requirements Engineering	Compliance Engineering	Reliability Engineering
Stakeholder identification, NWDs assessment, and reconciliation	Identify laws, regulations & standards	Define minimum necessary reliability
Risk management	Applicability assessment	Fault prevention
Design Inputs formulation & CAPA-driven changes	Design impact assessment	Fault removal/fault tolerance
Design Outputs engineering	Operational & salvage/disposal considerations	Fault/failure forecasting
Verifications & validation TEST DESIGN	Compliance TEST DESIGN	Reliability TEST DESIGN

Figure 7: HCSE Domain Activities

A. Stakeholder Identification & NWDs

A critical element of HCSE is the discovery and identification of all the stakeholders. Figure 8 shows some medical device stakeholders. Missing or misidentifying stakeholders will result in unexpected SD that will increase human error and undermine safety. Unfortunately, there is no guaranteed method of identifying all the stakeholders at any one point in time; it is for this reason HCSE, like SE, is an iterative engineering paradigm. Multiple iterations through the lifecycle permit the development/deployment teams to learn and refine their understanding of the intended use and the potential for error in the development/deployment design.

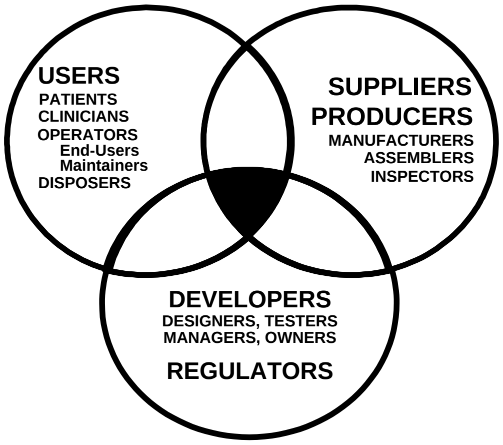

Figure 8: Some Medical Device Stakeholders

Human factors professionals (industrial engineers and psychologists) are trained to use a number of scientific techniques to help identify stakeholder NWDs. These include interviews, focus groups, questionnaires, and observational studies. These are not *ad hoc* endeavors, but scientifically-valid, statistically-controlled procedures for collecting and classifying stakeholder NWDs. The objective is not to find *some* of the NWDs; the objective is to find *all* of the NWDs for *all* of the stakeholders. What often limits reaching this objective is the perceived criticality of human error (e.g., can it kill or seriously injure someone?), as well as time and budget constraints. In every effort to develop and deploy a product, process, or service, we always are constrained to balance time, cost, scope, and quality. But, consider your own experience with any consumer product that provides a superb user experience versus a competitor's product that provides a mediocre user experience, think of your frequency of errors with one or the other, and then assess the relative success of the two competitors. Reducing the probability of use and user errors is a winning financial strategy.

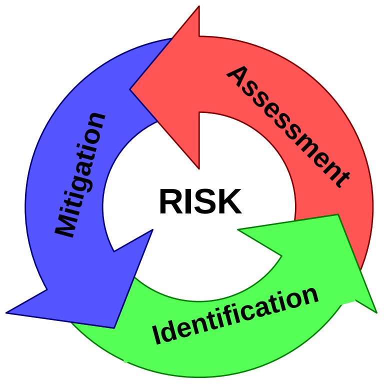

Figure 9: Risk Management

B. Risk Management

Risk management in human factors engineering is conducted exactly the same way as in hardware, software and economics engineering (see Figure 9). The same approach (inductive and deductive risk analysis) is used (Figure 10). The difference is focus; the focus in human factors engineering is on human injury and human error. One objective is to avoid acute & chronic health hazards and occupational safety hazards. Another objective is to identify, assess, and mitigate development-induced and organizationally-induced human errors.

Type of Input Data

Type of Risk Analysis		Quantitative, Historical	Subjective, Experiential
	Inductive (*Bottom Up*)	Failure Modes, Effects and Criticality Analysis (FMECA)	Failure Modes Effects Analysis (FMEA) Hazard & Operability Studies (HazOp) Hazard Analysis & Critical Control Points (HACCP)
	Deductive (*Top Down*)	Fault Tree Analysis (FTA) Event Tree Analysis (ETA)	Root Cause Analysis (RCA)

Figure 10: Objective vs Subjective Risk Analyses

An additional objective is to predict user (operator, maintainer, disposer, etc) errors (Figure 11), so that they may be identified, assessed, and mitigated either during development or deployment.

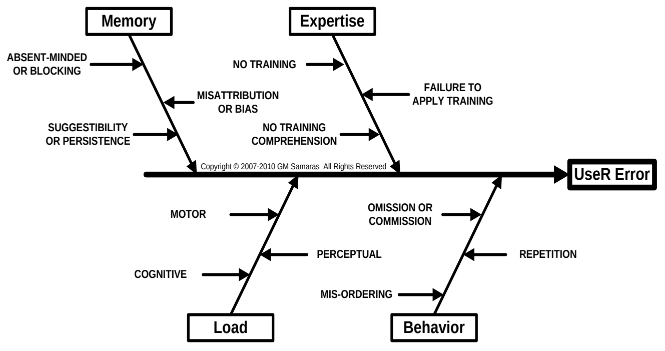
Memory
Expertise
ABSENT-MINDED OR BLOCKING
NO TRAINING
FAILURE TO APPLY TRAINING
MISATTRIBUTION OR BIAS
SUGGESTIBILITY OR PERSISTENCE
NO TRAINING COMPREHENSION
Copyright © 2007-2010 GM Samaras All Rights Reserved
UseR Error
MOTOR
OMISSION OR COMMISSION
PERCEPTUAL
REPETITION
COGNITIVE
MIS-ORDERING
Load
Behavior

Figure 11: Deductive user error analysis

As an engineering profession, we have learned that "tossing it over the wall" is unacceptable and we have instituted *concurrent engineering* to gain representation from manufacturing and service in the development process. We now need to learn that throwing it over the "next wall" (to the user organization where it will be deployed) is equally unacceptable. Merely accepting the product, process or service requirements from marketing no longer maintains competitiveness; what is now required is gaining representation from the deploying organizations, so that we can begin to address the risks resulting from the interaction of development and deployment decisions.

C. Formulating Design Inputs

Product, process, or service design inputs are some subset of identified stakeholders' NWDs; they are chosen because they are, at that point in time, economically and technologically feasible. In fact, design inputs comprise both requirements (what must be done) and constraints (what must not be done). An important function of the human factors engineer is to represent not only the end-users, but also the other identified stakeholders, helping the design team understand the impact of accepting or rejecting NWDs during the formulation of the design inputs. An equally important function of the human factors engineer is assisting in the operationalization of the design inputs. Operationalizing a design input means specifying exactly what must be measured and how it must be measured, so that the design can be validated (see Validations below).

D. Engineering Design Outputs

Just as with hardware, software, and economics engineering, human factors engineering proceeds (iteratively) in two phases: problem analysis and solution specification. For human factors engineering, this analysis begins with consideration of utility, esthetics, and individual differences. The analysis includes considering issues of acceptance and satisfaction, which typically are foreign to traditional engineering. Analytical tools include work domain analysis, function-task analysis, and cross-functional flow analyses; each of these is a technique for decomposing the problem into smaller, more manageable problems.

The human factors analysis considers human interface issues (see left side of Figure 5). One example is considering the size, feel, color and arrangements of physical controls and displays ("knobs & dials ergonomics"). Another example considers information management behaviors: mental workload issues, logic of operations issues, and potential training requirements for different design alternatives. A third example considers communication and coordination activities of various cooperating individuals within the work organization; these include standard operating procedures, work instructions, color-coding expectations, etc. A fourth example considers language and value system issues among coworkers: differences between clinicians and engineers in safety communications (e.g., meanings of specific terms used in warnings) or motivations among coworkers (e.g., clinicians desire for personal autonomy versus engineers desire for use of the latest technologies). Recommendations from human factors engineers to hardware or software colleagues often include design for 5%-95% population characteristics, the use of standardization (e.g., widely used display formats, familiar controls, adopting checklists, etc.) and forcing functions (engineering controls that act as interlocks).

E. Validations

Validation is the defense against an error of the third kind [5] – correctly solving the wrong problem. Validations include software validation, labeling comprehension validation, and system validation. Unlike verifications, validation requires the use of human subjects - typically intended users operating in an intended use environment (or a high fidelity simulation). As with stakeholder NWD assessment, human factors professionals can contribute to the design, implementation, and analysis of the validation studies.

There are important limitations to validation [7]. Validation is defeated if design inputs are improperly operationalized. Since validation is based solely on the documented design inputs, if a design input is absent (a latent failure), the tool will incorrectly pass validation. In the case of periodic revalidations after deployment (routinely required in pharmaceutical manufacturing), maintenance requirements that were not anticipated (another latent failure) or that deviate from the original design (a drift failure) may also result in the tool incorrectly passing re-validation.

IV. Discussion

HCSE extends SE, which is the fundamental engineering paradigm for hardware, software, and economics engineering. It incorporates human factors engineering throughout all phases of the lifecycle and emphasizes identification of all the stakeholders, assessing their NWDs, and attempting to reconcile conflicting NWDs. In this manner, it goes beyond traditional safety engineering to focus on defense against human errors in both development and deployment, instead of merely prevention of human injury.

References

[1] Akao, Y. “History of quality function deployment in Japan” (1990, vol. 3, pp. 183-196), International Academy for Quality: IAQ Book Series.

[2] Buede DM. The Engineering Design Of Systems: Models And Methods New York: Wiley, 2000, pg 6.

[3] Dekker SWA. Ten Questions about Human Error: A New View of Human Factors and System Safety, New Jersey: Lawrence Erlbaum Associates, 2005

[4] Kano, N. “Attractive quality and must-be quality”, Journal of the Japanese Society for Quality Control, April 1984, *14*(2), 39-48.

[5] Mosteller F. “A k-sample slippage test for an extreme population”. Ann Math Stat. 1948, 19(1): 58-65

[6] Reason J. Human Error. Cambridge: Cambridge University Press. 1990

[7] Samaras GM. “An Approach to Human Factors Validation”. J. Validation Technology, 2006, 12(3): 190-201

[8] Samaras GM. “Human-Centered Systems Engineering: Building products, processes, and services”. Proc 2010 SHS/ASQ Joint Conference. February 26-28. Atlanta, GA (CD-ROM)

[9] Samaras, GM. The Use, Misuse, and Abuse of Design Controls. IEEE Eng Med Biol Magazine 29(3):12-18, 2010

[10] Samaras GM. “Human-Centered Systems Engineering: Managing Dissonance in Healthcare Delivery”, in Management Engineering for Effective Healthcare Delivery: Principles and Practices, Kolker, A. & Story, P. (Eds). Philadelphia: IGI Global 2011, pg. 148-171.

[11] Samaras GM & Horst RL. “A systems engineering perspective on the human-centered design of health information systems”. J Biomedical Informatics. 2005, 38(1):61-74.

[12] Samaras GM & Samaras EA. (2009). Feasibility of an e-Health Initiative: Information NWDs of Cancer Survivor Stakeholders. Proc. IEA 2009. Beijing, China. August 9-14.

[13] Samaras, E. A. & Samaras**,** G. M. “Using Human-Centered Systems Engineering to Reduce Nurse Stakeholder Dissonance”. Biomedical Instrumentation & Technology, 2011, 44(s1):25-32

[14] Simon, H. A. Models of man: Social and rational. New York, NY: Wiley, 1957

[15] Woods DD & Hollnagel E. “Prologue: Resilience Engineering Concepts in Resilience Engineering”, Hollnagel, Woods, & Leveson (Eds) Hampshire, England: Ashgate, 2006, pg 3.