Human-Centered Systems Engineering: Building Products, Processes, and Services

GM Samaras, Samaras & Associates, Inc.
Pueblo, CO, USA

Abstract

Products, processes, and services exist solely because their use by humans has real or perceived value (utilitarian or esthetic). Introducing humans into systems dramatically increases system complexity. An approach to human-centered system complexity (from physical and behavioral to social and cultural considerations) encourages appreciating the system interfaces both to individual humans and their organizations.

Systems Engineering offers a structured, systematic approach to the conceptualization, design, development, deployment, and replacement of products, processes, and services. In existence since the early 1900s, it is characterized by a state space and visualized as a lifecycle. Human-Centered Systems Engineering is an extension emphasizing the criticality of human actors, and their organizations, in the engineering process. Ignoring these interfaces to the system results in various types of errors, including Reason's latent flaws and Dekker's drift.

For human-centered systems engineering, quality is about identifying and *satisficing* ALL the stakeholders' evolving and frequently conflicting Needs, Wants, and Desires. A human-centered approach presents a rather large set of factors for engineering verification and validation studies. Experimental design approaches historically used by engineers are very inefficient given large numbers of factors. Statistical design of experiments and variations, such as those of Taguchi and Shainin, offer a more economical approach for dealing with the many variables that is arise in human-centered product, process, and/or service verification and validation studies.

Human Actors & System Complexity

Products, processes, and services are developed and maintained solely because their use by humans has real or perceived value (utilitarian and/or esthetic). Even completely automated, unsupervised systems have human users – maintenance personnel; maintenance is typically a significant portion of the total cost of ownership. This is the fundamental justification and rationale for human-centered systems engineering (HCSE). Historically, system designers have viewed human operators as unreliable and inefficient; they strive to supplant them with automation or ignore them. As Bainbridge (1983) pointed out, designer's errors are significant contributors to accidents and undesirable events. The irony is they still leave to humans the tasks that the designer cannot think how to automate.

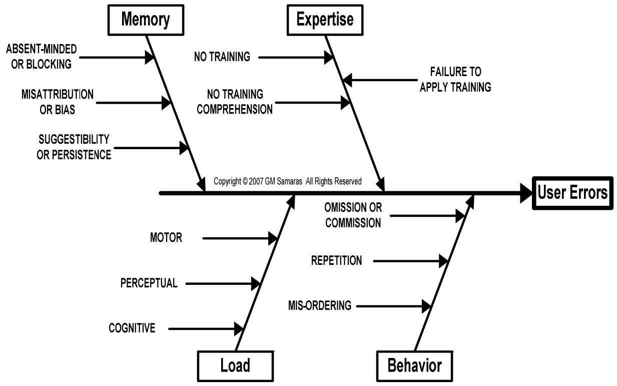

Memory
Expertise
ABSENT-MINDED OR BLOCKING
MISATTRIBUTION OR BIAS
SUGGESTIBILITY OR PERSISTENCE
NO TRAINING
NO TRAINING COMPREHENSION
FAILURE TO APPLY TRAINING
Copyright © 2007 GM Samaras All Rights Reserved
User Errors
MOTOR
PERCEPTUAL
COGNITIVE
OMISSION OR COMMISSION
REPETITION
MIS-ORDERING
Load
Behavior

Figure 1: UseR Errors Root Cause Analysis (partial)

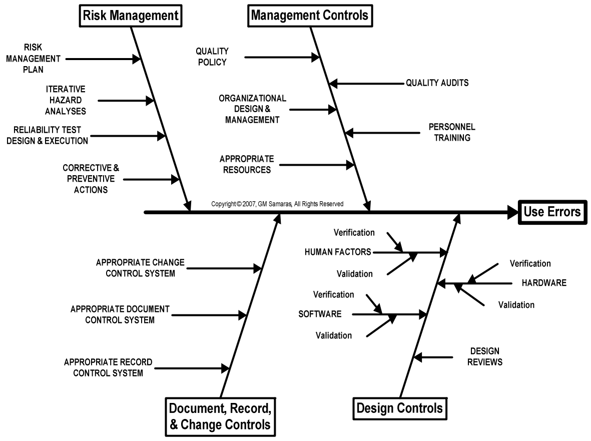

Risk Management
Management Controls
RISK MANAGEMENT PLAN
ITERATIVE HAZARD ANALYSES
RELIABILITY TEST DESIGN & EXECUTION
CORRECTIVE & PREVENTIVE ACTIONS
QUALITY POLICY
QUALITY AUDITS
ORGANIZATIONAL DESIGN & MANAGEMENT
PERSONNEL TRAINING
APPROPRIATE RESOURCES
Copyright © 2007, GM Samaras, All Rights Reserved
Use Errors
APPROPRIATE CHANGE CONTROL SYSTEM
APPROPRIATE DOCUMENT CONTROL SYSTEM
APPROPRIATE RECORD CONTROL SYSTEM
Verification
HUMAN FACTORS
Validation
Verification
HARDWARE
Validation
Verification
SOFTWARE
Validation
DESIGN REVIEWS
Document, Record, & Change Controls
Design Controls

Figure 2: Use Error Root Cause Analysis (partial)

Introducing human actors into any endeavor dramatically increases the possible number of incorrect or inappropriate responses of a "simple" system. The ratio of "wrong to right" responses often is used to characterize the complexity of tasks; it also is used to impute the requisite level of expertise (training and experience) of the user (or groups of users and/or automated aids) to execute successfully a series of such tasks. We generally consider four types of human error: *Use*, unexpected *Use*, mis*Use*, and ab*Use*. However, this ignores the difference between two general categories of human error: UseR errors (Figure 1) are attributable to the internal or external user environment, excluding the system itself; Use errors (Figure 2) are attributable to the design and/or implementation of the system.

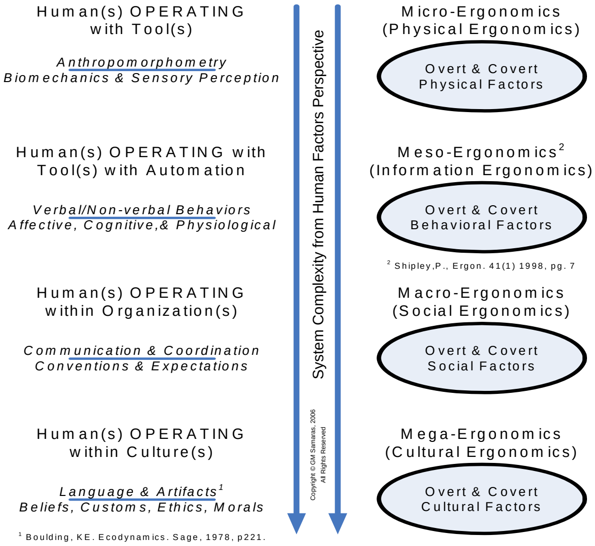

Human(s) OPERATING with Tool(s)
Anthropomorphometry
Biomechanics & Sensory Perception
Human(s) OPERATING with Tool(s) with Automation
Verbal/Non-verbal Behaviors
Affective, Cognitive,& Physiological
Human(s) OPERATING within Organization(s)
Communication & Coordination
Conventions & Expectations
Human(s) OPERATING within Culture(s)
Language & Artifacts[1]
Beliefs, Customs, Ethics, Morals
[1] Boulding, KE. Ecodynamics. Sage, 1978, p221.
System Complexity from Human Factors Perspective
Copyright © GM Samaras, 2006
All Rights Reserved
Micro-Ergonomics
(Physical Ergonomics)
Overt & Covert
Physical Factors
Meso-Ergonomics[2]
(Information Ergonomics)
Overt & Covert
Behavioral Factors
[2] Shipley,P., Ergon. 41(1) 1998, pg. 7
Macro-Ergonomics
(Social Ergonomics)
Overt & Covert
Social Factors
Mega-Ergonomics
(Cultural Ergonomics)
Overt & Covert
Cultural Factors

Figure 3: Human-Centered System Complexity

Humans dramatically increase system complexity. Complex systems have *emergent* properties – the result of component interactions – that are not readily predictable without appreciation of the system as a whole. It is now generally recognized that design-induced errors are a serious problem, a critical system safety issue, and an important source of reduced quality; they cannot be alleviated with just user training! Not fully appreciating human-centered system complexity, especially in project risk analysis, has been an important obstacle in the design and implementation of essential systems (e.g., see how a decade's difference dramatically altered perspectives of computerized physician order entry [Tierney et al, 1993 vs. Koppel et al, 2005]).

A human-centered approach requires that we must achieve a detailed appreciation for the interfaces to human actors as well as the interfaces between actors; otherwise, we remain unable to predict and control the critical human/organizational influences both on system design parameters and on system sensitivities to external factors. Our fundamental need to study the system as a whole leads to a model of human-centered system complexity (Figure 3), one way of appreciating both the system components and their potential interactions. Of the four levels in the model, the first two levels identify attributes of the interfaces with individual actors and their tools; the last two levels address the attributes of the interfaces between groups of actors (see also Figure 11). In all cases, they allow us to operationalize (and thus measure) the overt and covert interface attributes. These four levels focus our attention on physical "size and fit", information dependent behaviors, social, and cultural considerations. It is in this last level (cultural ergonomics) that we encounter a structure to focus, *for example*, on the assumptions and critical differences in language, tools, and customs between an engineering subculture (developing a system) and a clinical subculture (using a system). This complexity model helps support comprehensive consideration of system, parameter, and tolerance design for engineering human-centered systems.

HCSE Fundamentals

Systems engineering (SE) is a structured, systematic approach to the development, deployment, and replacement of products, processes, and services. HCSE extends SE to emphasize the criticality of human actors <u>and</u> their organizations in the engineering process. The state space for tools (Figure 4) identifies the range, domain, and timeline of engineering activities (S/P: seller/purchaser; RDDT&E: research, design, development, testing, & evaluation); it does not clarify the essential iterative nature of the SE process (Figure 5).

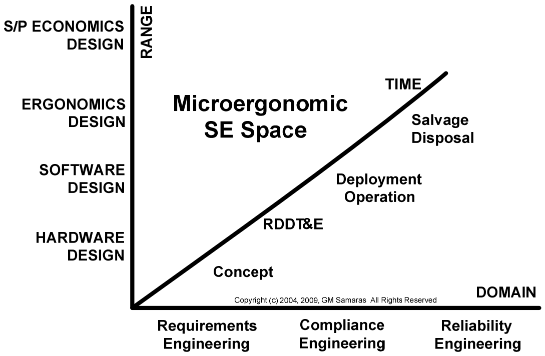


Figure 4: HCSE State Space

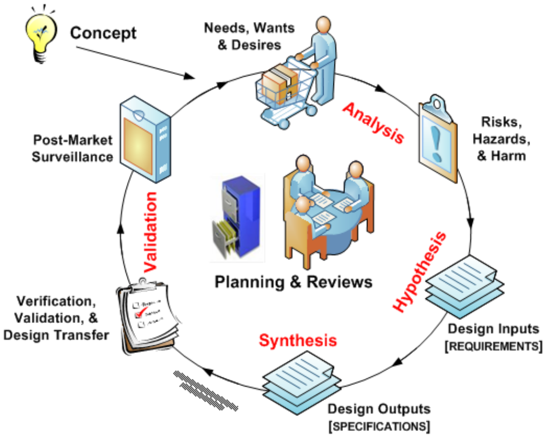


Figure 5: Iterative Development Paradigm

Ignoring or misapplying SE principles and practices results in the emergence of system errors, most of which lead to undesirable events. There are two general categories, both the result of inadequate engineering management: *propagated* errors and *compounded* errors. Propagated errors (Figure 6) permit errors and omissions made early in the design cycle to continue uncorrected through to system implementation and deployment; they can be effectively managed with proper engineering risk management, verification, and engineering validation studies.

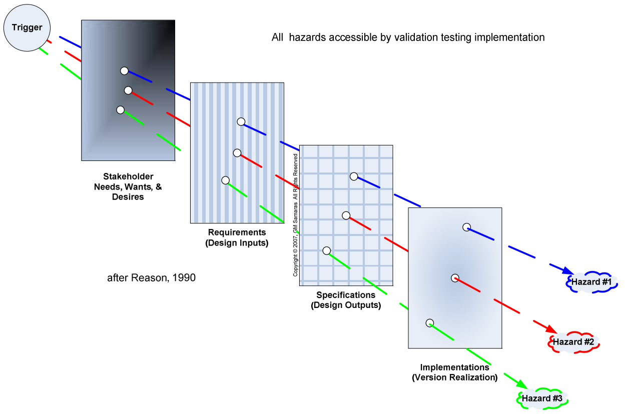

Trigger
All hazards accessible by validation testing implementation
Stakeholder Needs, Wants, & Desires
Requirements (Design Inputs)
Specifications (Design Outputs)
Implementations (Version Realization)
Hazard #1
Hazard #2
Hazard #3
after Reason, 1990
Copyright © 2007, GM Samaras All Rights Reserved

Figure 6: Illustration of Propagated Errors

Compounded errors (Figure 7) cannot and demonstrate the limitations of engineering validation; two are illustrated: latent flaws (Reason, 1990) and drift (Dekker, 2005). The *latent flaw* (Figure 7: Hazard #1) is the result of an incorrect Design Input masked by a Design Output defect. The specification *drift* (Figure 7: Hazard #3) is the result of an incorrect Design Output masked by an Implementation defect.

In both cases, intentional or inadvertent correction of the defect causes the hazard "suddenly" to become an *unanticipated* system failure. These errors can occur for hardware, software, human factors, and seller/purchaser economics engineering as well as their combinations. It seems the only defense currently known is draconian application of complete and correct systems engineering principles and practices.

One well-established source of such errors is confusing or intermingling Design Inputs (Requirements; what we agree to build) with Design Outputs (Specifications; the engineers' work product). This virtually always guarantees a sub-optimal engineering solution and the existence of propagated and/or compounded errors. The "rule of thumb" for discriminating between the two: if it has, or should have, a numerical value, physical units, and a tolerance, it is most likely a Specification; if not, it is probably a Requirement. The sources of Design Outputs are one or more Design Inputs; the sources of Design Inputs are the Stakeholders' Needs, Wants, and Desires (NWDs). Design Inputs are a selected subset of NWDs that are deemed technically and economically feasible.

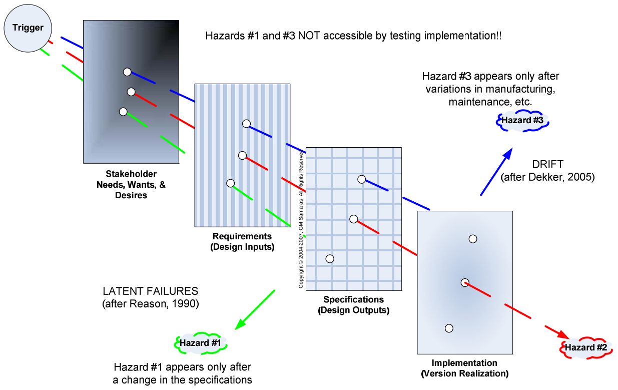

Trigger
Hazards #1 and #3 NOT accessible by testing implementation!!
Hazard #3 appears only after variations in manufacturing, maintenance, etc.
Hazard #3
DRIFT (after Dekker, 2005)
Stakeholder Needs, Wants, & Desires
Requirements (Design Inputs)
Specifications (Design Outputs)
Implementation (Version Realization)
LATENT FAILURES (after Reason, 1990)
Hazard #1
Hazard #1 appears only after a change in the specifications
Hazard #2
Copyright © 2004-2007, GM Samaras All Rights Reserved

Figure 7: Illustration of Compounded Errors

Defining HCSE Quality

HCSE quality may be defined as *the degree to which the system satisfices the NWDs of all the stakeholders*. From the work of Kano (1984), we have a simple means of discriminating NWDs (Figure 8). Simon (1957) coined the term "satisfice"; it means obtaining a good result that is good enough, although not necessarily the best, for each of the stakeholders. This describes a nonlinear programming problem well known in the Operations Research field (not to be confused with simple engineering tradeoffs). In applying HCSE, difficulty arises in both the identification of stakeholders and the non-alignment of their often conflicting and evolving NWDs (Figure 9).

	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 8: Stakeholder Response Matrix

In HCSE, the emphasis shifts to iterative discovery of stakeholders, identifying their evolving NWDs, and reconciling the conflicts, to satisfice the whole group (concurrent engineering is a subset of this approach). The shift in emphasis tends to mitigate errors and omissions early in the system development cycle, reducing their final cost. Absent robust HCSE, essential systems will continue to hinder rather than help, be economically inefficient, and be examples of poor quality. However, to manage this, we must be able to measure and control the interface attributes.

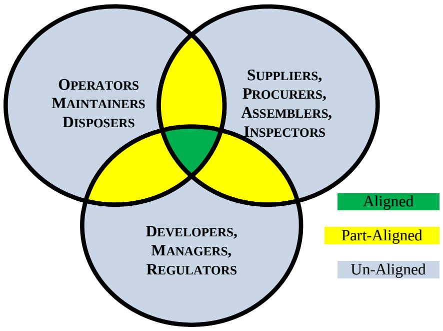


Figure 9: Alignment of Stakeholder NWDs

HCSE Metrology Issues

Stakeholders operate in a complicated environment (Figure 10) that influences what they achieve and how they err. We can recast the complexity model (Figure 3) as a table of measurement categories (Figure 11). Examination of each category identifies that the metrology belongs to a variety of scientific disciplines – from biomechanics to cultural anthropology. Physical measurements include essentially static human characteristics as well as dynamic measurements used in biomechanics. Behavioral measurements use traditional techniques of experimental psychology. Techniques of social anthropology, social psychology, and sociology are used for social measurements. Cultural measurements use techniques of linguistics (language), archaeology (tools and other artifacts), and cultural anthropology (value systems).

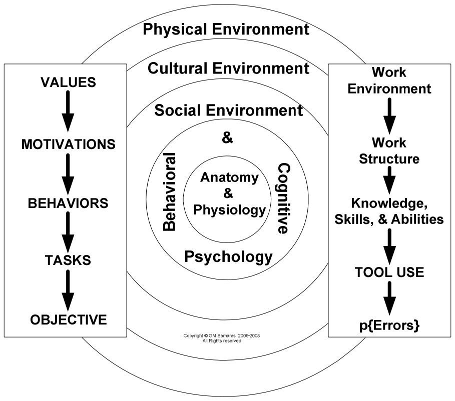


Figure 10: Factors for Actors

Therefore, if someone tells you "*all that human-centric stuff is well and good, but I cannot operationalize and measure it*" respond that they are not expected to – it takes an interdisciplinary team (but we must also recognize there will be attendant engineering management problems). There may be an enormous number of variables involved, some important or even critical and others that may have little or no impact on your system. However, you cannot manage what you cannot control and you cannot control what you cannot measure. One of the challenges of implementing HCSE is dealing with the many measurement variables; it is not that big of a challenge.

	METROLOGY			
	INDIVIDUALS		GROUPS	
	PHYSICAL	BEHAVIORAL	SOCIAL	CULTURAL
OVERT	ANTHROPO- MORPHO- METRY	VERBAL & NON-VERBAL	COMMUNICATION & COORDINATION	LANGUAGE & ARTIFACTS
COVERT	BIOMECHANICS & SENSORY PRECEPTION	AFFECTIVE COGNITIVE PHYSIOLOGICAL	CONVENTIONS & EXPECTATIONS	BELIEFS CUSTOMS ETHICS MORALS

Figure 11: HCSE Metrology Categories

Managing Many Factors

System designers typically think in terms of experimental designs (for engineering validation studies) that alter one variable at a time (OVAT). Such designs are typically not economical (especially in the presence of very large numbers of measurement variables), they do not readily support identification of optimality conditions, and they provide little information on sources of variability. This recognition has motivated designers to shy away from methods producing large numbers of measurement variables!

At the beginning of the 20th century, Fisher (1926) conceived of a statistically rigorous and universal framework to design and analyze all comparative experiments. It permits simultaneous study of individual and interactive effects of multiple variables … and it has a very simple underlying geometric structure. It will support identification of optimality conditions and provide information on sources of variability. The workhorse of Design of Experiments (DOE) is the full factorial design; it is a design in which every setting of every variable appears with every other setting of every other variable. To visualize the geometry (Figure 12), consider a design with five variables (factors); the full factorial design yields 32 $(n^k, \forall\, n = 2, k = 5)$ runs (the solid black vertices), but only if the assumption is made that you are dealing with linear variables $(n = 2)$. While very powerful, such designs get very big very fast; for nonlinear variables, a quadratic assumption requires $n = 3$ and a cubic assumption requires $n = 4$.

To avoid full factorial designs for large numbers of variables ("task overload"), fractional factorial designs are employed. Fractional designs reduce your workload by explicitly assuming 3rd and higher order interaction effects are not important; they only identify main effects (effect of changing one variable alone on the system response) and 2nd order effects (effect of changing one variable on another variable's effect on the system response). One famous example is the Taguchi (1987) simplification (Figure 13); retaining the linearity assumption, but recognizing that some variables are independent of others (e.g., controlled variables versus noise variables), the 32-run design reduces to a more manageable, more economical 8-run design (the light grey vertices). There exist a large number of other fractional factorial designs; they may be readily accessed and implemented using commercial statistical software.

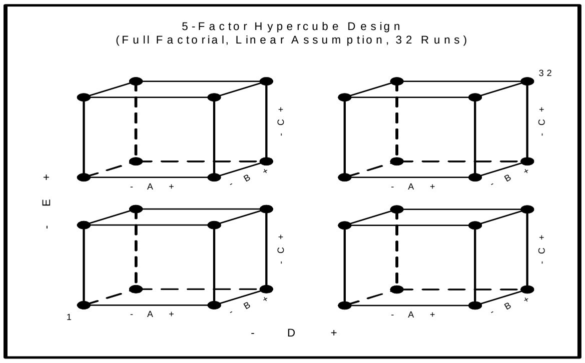

5-Factor Hypercube Design
(Full Factorial, Linear Assumption, 32 Runs)
32
- C +
- A +
- B +
- E +
1
- D +

Figure 12: Full Factorial Design

An approach different from the fractional factorial designs is that of Shainin (see Bhote, 1988; Anthony & Cheng, 2003); it uses a Variables Search approach to identify the critical variables out of a large number of candidates (conceptually, application of the "Pareto Principle" or "Juran Assumption"). After reducing the number of critical variables to less than five, the approach employs a traditional full factorial design. Shainin's approach goes on to validate the findings and then to optimize the results – using only mathematical methods known to high school students.

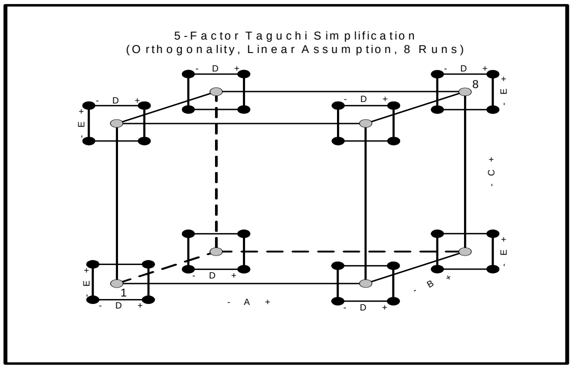
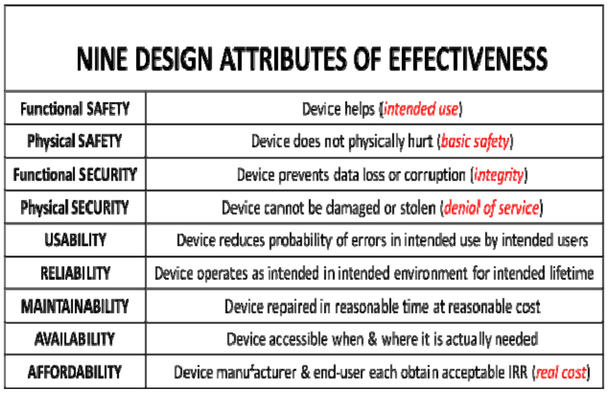

5-Factor Taguchi Simplification
(Orthogonality, Linear Assumption, 8 Runs)
- D +
- E +
8
- C +
1
- A +
- B +

Figure 13: Taguchi Simplification

Why is it important to consider DOE and its variations for experimental design? Until the System Implementation has been experimentally validated against the original Requirements, there is no assurance that the correct system has been built. The nine design attributes shown in Figure 14 are general statements of various stakeholder NWDs, not design inputs or outputs. They need to be expressed more specifically and operationalized before becoming Design Inputs. Once operationalized, they also become the basis for engineering validation.



NINE DESIGN ATTRIBUTES OF EFFECTIVENESS	
Functional SAFETY	Device helps (*intended use*)
Physical SAFETY	Device does not physically hurt (*basic safety*)
Functional SECURITY	Device prevents data loss or corruption (*integrity*)
Physical SECURITY	Device cannot be damaged or stolen (*denial of service*)
USABILITY	Device reduces probability of errors in intended use by intended users
RELIABILITY	Device operates as intended in intended environment for intended lifetime
MAINTAINABILITY	Device repaired in reasonable time at reasonable cost
AVAILABILITY	Device accessible when & where it is actually needed
AFFORDABILITY	Device manufacturer & end-user each obtain acceptable IRR (*real cost*)

Figure 14: Medical Device Design Attributes

Conclusion

Many incidents and accidents are alleged to be human error, but we must remember that there are both Use errors and UseR errors. Human Use errors of the system are, in large part, within the locus of control of system developers. Even future UseR errors can be influenced by the developer (e.g., avoid confusing or frustrating the operator, avoid undesirable physical or cognitive exercises, avoid delays and operator attention loss). In the health care arena, safe and effective systems (products, processes, and services) are desired; they are the system developer's goal. However, human stakeholders complicate the engineering process at a myriad of levels from conceptualization through development, deployment, and replacement. It is only recently that there has been a concerted effort to include systematic consideration of human factors and ergonomics (HF/E). However, HF/E is still the "new kid on the block" and is treated as "specialty" engineering, when, in fact, everything else but HF/E should be considered specialty engineering – after all, the ultimate purpose for building and fielding the system is human use.

HCSE is an extension of classical SE and is an attempt to integrate HF/E considerations throughout the system lifecycle – from "lust to dust". It is an attempt to consider the full range of human issues (physical, behavioral, social, and cultural) in a systematic manner that leverages the measurement capabilities of a wide variety of scientific disciplines, many of which are only now being considered useful for system development.

References

Anthony J & Cheng AHY. (2003). Training for Shainin's approach to experimental design using a catapult. J. Eur. Indust. Training. 27(8): 405-412.

Bainbridge L. (1983). Ironies of automation. Automatica, 19.775-779. Retrieved January 13, 2010 from:
http://www.bainbrdg.demon.co.uk/Papers/Ironies.html

Bhote KR. (1988). "World Class Quality: Design of Experiments Made Easier, More Cost Effective than SPC". New York: AMA Membership Publication Division.

Dekker SWA. (2005). "Ten Questions about Human Error: A New View of Human Factors and System Safety". New Jersey: Lawrence Erlbaum Associates

Fisher RA. (1926). The arrangement of Field Experiments. J. Ministry Agri. Great Britain. 33:503-513. Retrieved January 13, 2010 from:
http://digital.library.adelaide.edu.au/coll/special//fisher/48.pdf

Kano N. (1984). Attractive quality and must-be quality. J. Japanese Soc. Qual. Control. April pp. 39-48.

Koppel R et al. (2005). Role of computerized physician order entry systems in facilitating medication errors. JAMA 293(10): 1197-1203.

Reason J. (1990). "Human Error". Cambridge: Cambridge University Press.

Simon HA (1957) "Models of Man: Social and Rational" New York: Wiley.

Taguchi G & Konishi S. (1987), Orthogonal Arrays and Linear Graphs, Dearborn, MI, ASI press.

Tierney WM, Miller ME, Overhage JM & McDonald CJ. (1993). Physician inpatient order writing on microcomputer workstations. Effects on resource utilization. JAMA 269(3): 379- 383.

Biographical Sketch

GM Samaras is a scientist & engineer in private practice. His firm provides technical, regulatory, and management consulting services primarily to medical device and pharmaceutical manufacturers. He has a number of biomedical engineering patents and numerous peer-reviewed publications in physiology and hardware, software, human factors, and quality engineering.

He is a licensed professional engineer (PE), an ASQ-certified quality engineer (CQE) and a board-certified professional ergonomist (CPE). He is trained as an electrical engineer and earned doctorates from the U. of Maryland (PhD 1976; Physiology & Biopsychology) and The George Washington U. (DSc 1992; Engineering Management & I/O Psychology).

His past employment includes serving as a medical school professor (U of MD, Baltimore) and engineering graduate school professor (GWU SEAS, Wash., DC), founding and managing a biomedical engineering firm, and working for the FDA/CDRH as a reviewer and manager.

Contact: george@samaras.eng.pro