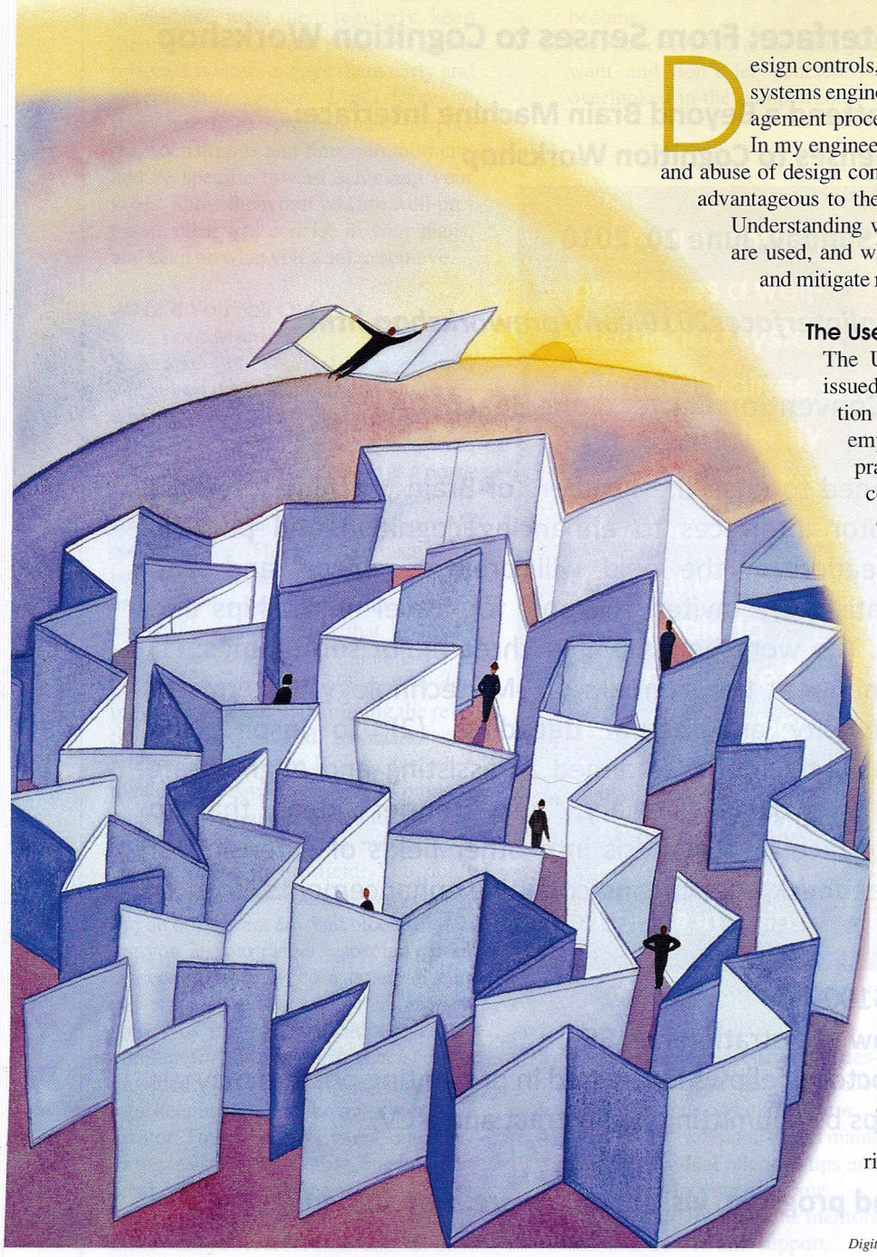
The Use, Misuse, and Abuse of Design Controls

BY GEORGE M. SAMARAS

Human-Centered Systems Engineering



© STOCKBYTE

Design controls, a relatively new name for a nearly century-old systems engineering paradigm, describe an engineering management process that serves both producers and consumers. In my engineering practice, I have observed the use, misuse, and abuse of design controls. Misuse and abuse are not economically advantageous to the producer and create risks for the consumer. Understanding what are design controls, how and when they are used, and why they add value may encourage proper use and mitigate risks for both producers and consumers.

The Use, Misuse, and Abuse of Design Controls

The U.S. Food and Drug Administration (FDA) issued a final rule called the *quality system* regulation in the Fall of 1996. Although the primary emphasis was a recast of good manufacturing practices regulations for medical devices, it contained a unique section called *design controls*. In my engineering practice since 1996, I have observed three fundamental approaches to design control compliance: 1) continuous improvement understanding and following the process, 2) reinterpretation of the terminology to conform to existing internal product development practices, and 3) a posteriori creation of the requisite design documentation. The latter two, misuse and abuse of design controls, are not economically advantageous; they do little to improve the attributes of effectiveness of the product (Table 1) and ultimately result in a reduced internal rate of return (IRR) for both the producer and the consumer. My personal observation is that those who misuse or abuse design controls during development of their medical products seem to have a much higher incidence of expensive product recalls and plaintiff litigation. Understanding exactly what are design controls, how and when they are used, and why they add value may mitigate risks for both producers and consumers.

Digital Object Identifier 10.1109/MEMB.2010.936551

 0739-5175/10/$26.00©2010IEEE

Design inputs are the subset of the stakeholders NWDs that the designing organization believes is technologically and economically feasible.

What Are Design Controls

The origins of engineering design controls (though not the name) trace back to classical systems engineering [1]. They are identified by both FDA's CFR Part 820.30 [2] and by international consensus standard ISO 13485 §7.3 [3]. The correct application of engineering design controls reduces the risks for both producers and consumers, decreases time to market for viable products, and satisfices identified stakeholders' needs, wants, and (often) desires (NWDs). (Needs are what each stakeholder believes they must have. Wants are what each stakeholder believes they would like to have. Desires, also called *latent needs*, are not known in advance by the stakeholders, but they know it when they see it.) *Satisfice*, a term coined by Simon [4], means to obtain a good result that is good enough, though not necessarily the best, for each stakeholder. The rationale is that different stakeholder groups have evolving and conflicting NWDs; these stakeholder dissonances must be reconciled. Design controls are processes that ensure a bottom-up product development approach (satisficing stakeholder NWDs) and not a top-down approach (finding a new intended use for a given technology).

Design controls may be applied to the development of products, processes, or services. For a medical device, Figure 1 shows the state space—a three-dimensional microergonomic systems engineering state space—that incorporates all possible engineering tasks throughout a tool's full life cycle from concept to disposal [1]. The domain of all these activities (see Table 2) consists of requirements engineering (what to build), compliance engineering (what not to build), and reliability engineering (reducing risks for both producer and consumer). [Compliance engineering may be viewed as what not to do: do not have accessible sharp points near wiring harnesses, do not have exposed high-voltage conductors or connection points, do not have product emit nonessential electromagnetic levels beyond a certain wattage, etc.] The range of these design engineering activities is hardware design, software design, human factors design, and seller/purchaser (S/P) economics design (the latter two being industrial engineering activities); each of these is characterized by multiple subdisciplines. Figure 2 shows the principal elements of the iterative design control process contrasted with the four elements of the scientific method.

In Figure 2, the initial task following conceptualization, and in each subsequent iteration, is identification of all the stakeholders and their NWDs. After reconciliation, a subset of these NWDs deemed technologically and economically feasible is chosen for formulation as design inputs. These design inputs are the problems presented to the technical staff; their solution (their work products) are the design outputs. In the transformation from reconciled stakeholders' NWDs to design outputs, a number of additional engineering activities take place, including development planning, risk management, five types of verifications, and

Table 1. 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 use environment for intended lifetime
Maintainability	Device repaired in reasonable time at reasonable cost
Availability	Device accessible when and where it is actually needed
Affordability	Device manufacturer and end user each obtain acceptable IRR (real cost)

IRR: internal rate of return.

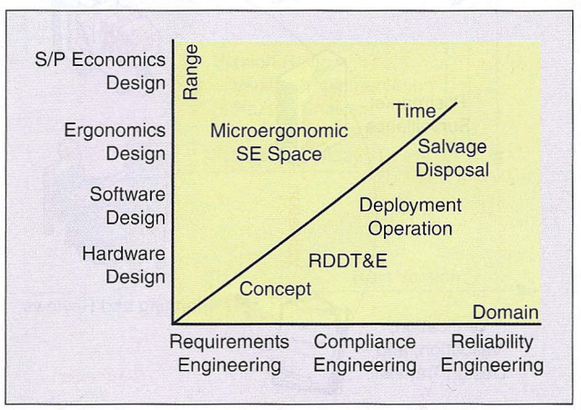


Fig. 1. The systems engineering state space for developing and deploying tools; all possible engineering activities reside on this three-dimensional framework (manufacturing and distribution are included in deployment and service is included in operation). S/P: seller/purchaser; RDDT&E: research, design, development, testing, and evaluation; SE: systems engineering.

Table 2. Activities in the systems engineering state space domains.

Requirements Engineering	Compliance Engineering	Reliability Engineering
Stakeholder identification, NWD assessment and reconciliation	Identification of laws, regulations, and standards	Defining minimum necessary reliability
Risk management	Applicability assessment	Fault prevention
Design input formulation and five verifications	Design impact assessment	Fault removal
Version validation	Test design	Fault tolerance
Version postmarket surveillance	Operational considerations	Fault/failure forecasting
CAPA-driven design input changes	Salvage and/or disposal considerations	Test design

NWD: needs, wants, and desires of stakeholders; CAPA, corrective and preventive action.

design reviews. They are not depicted here in the actual sequence that they occur; please refer to Figure 3 for the detailed flowchart. With each iteration, engineering validation experimentally demonstrates that design inputs were or were not correctly translated to the current implementation. On the last iteration, the final design outputs are transferred to manufacturing for mass production. Postmarket surveillance helps identify missing misunderstood NWDs that allow future corrective and preventive actions.

The FDA-mandated design controls regulation (21 CFR 820.30:1996) changed two key terms (requirements and specifications) that were historically terms of art in classical systems engineering; these are now called design inputs and design outputs, respectively. Design inputs are the subset of the stakeholders NWDs that the designing organization believes is technologically and economically feasible; they are testable natural language (e.g., English) statements understandable by all stakeholders. Design outputs are how engineers solve the problem posed by the design inputs; often, there may be multiple solutions, depending upon what optimization criteria are applied. Design outputs (the design engineers' work product) tell manufacturers what to build and how to build it. These modern terms resolve a longstanding nomenclature problem that occurs when English-speaking engineers and managers interchange systems engineering terms with lay terms, creating phrases that are technically ambiguous or nonsensical (e.g., requirements specification and specification requirements). With the modern terms, it is now possible to speak unambiguously of a design inputs specification (a documented compendium of design inputs, also called a design requirements document) and design output requirements (the required elements of a properly formulated design outputs document). Often, it is confusing trying to identify whether something is a requirement or a specification. A useful rule of thumb is if it has a value, unit, and tolerance, it's a spec; if it doesn't, it's probably a requirement.

Design controls are nothing more than the fundamental elements of classical systems engineering. The classical systems engineering process is a very powerful risk-reduction mechanism that serves both consumers and producers, but whose value is diluted or negated if not followed rigorously. Three key process attributes for correctly applying design controls are: 1) multiple iteration, 2) comprehensive contemporaneous documentation, and 3) flexible decision making. The human focus, iteratively reassessing stakeholders and reconciling their often conflicting and evolving NWDs, moves the innovation process toward human-centered systems engineering [5].

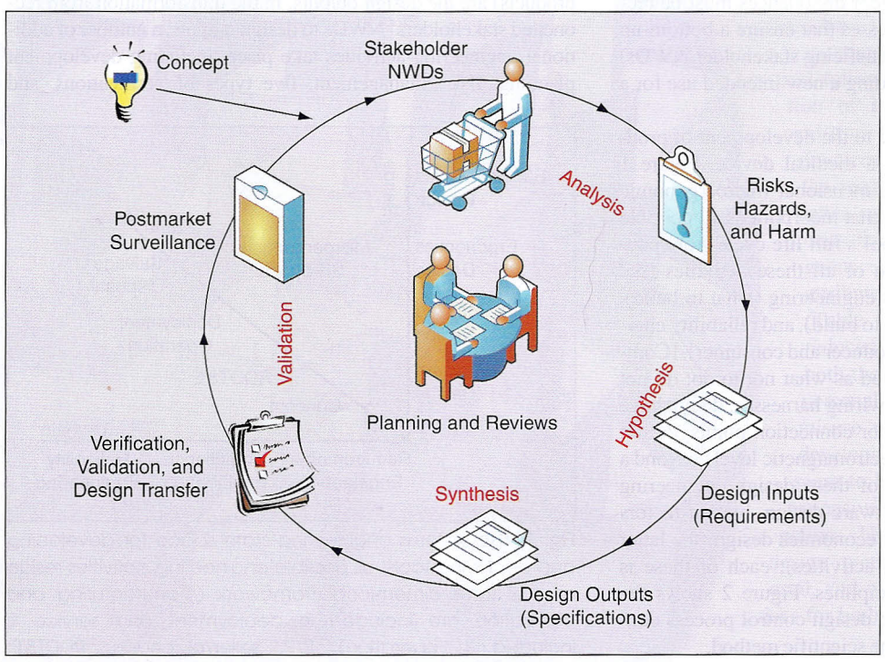


Fig. 2. The major elements of design controls are mapped on the four elements of the scientific method.

Multiple Iteration

Developing new/improved products is always a learning process; iteration allows the engineering team to engage in structured learning, while time-constrained iterations may make the process agile. It has always been unrealistic to believe that first you establish

the requirements, then you develop the specifications, test the implementation, and field the product. The waterfall model [6], [7], except for the most trivial innovations, is never strictly implemented in the real world. Iteration always occurs, even though it may be disguised or denied. The human learning process is incremental and builds upon experience and repetition, primarily gained from testing—some of which is formal, but much of which is informal, accidental, and experiential. Figure 3 illustrates the iterative process in the form of a flowchart. Each iteration during product development yields an internal release, until the final version yields the external release. The multiple feedback loops formed by verification testing (of design inputs formulation, design outputs development, and design outputs implementation) provide the opportunity to correct internal misunderstandings and technical errors. Two additional types of verification are not illustrated here: verification that risk mitigation was properly applied and verification that properly applied mitigation actually reduced risk [8, paragraph 6.3]. The validation loop (from implementation back to design inputs) permits identification and correction of a mismatch between what was agreed would be built and what was actually built; this is invariably a clinical (clinical refers to dealing with humans, patients in the case of medicine and psychology, and users in the case of human factors engineering) trial involving users in their expected use environment [9]. A corrective and preventive action loop is the path to the next iteration; it provides the opportunity to correct major and minor flaws, most often related to internal or external complaints about the identified stakeholders' evolving NWDs. Iteration alone, without rigorously following the process in Figure 3, is inadequate. Understanding what will satisfice all the

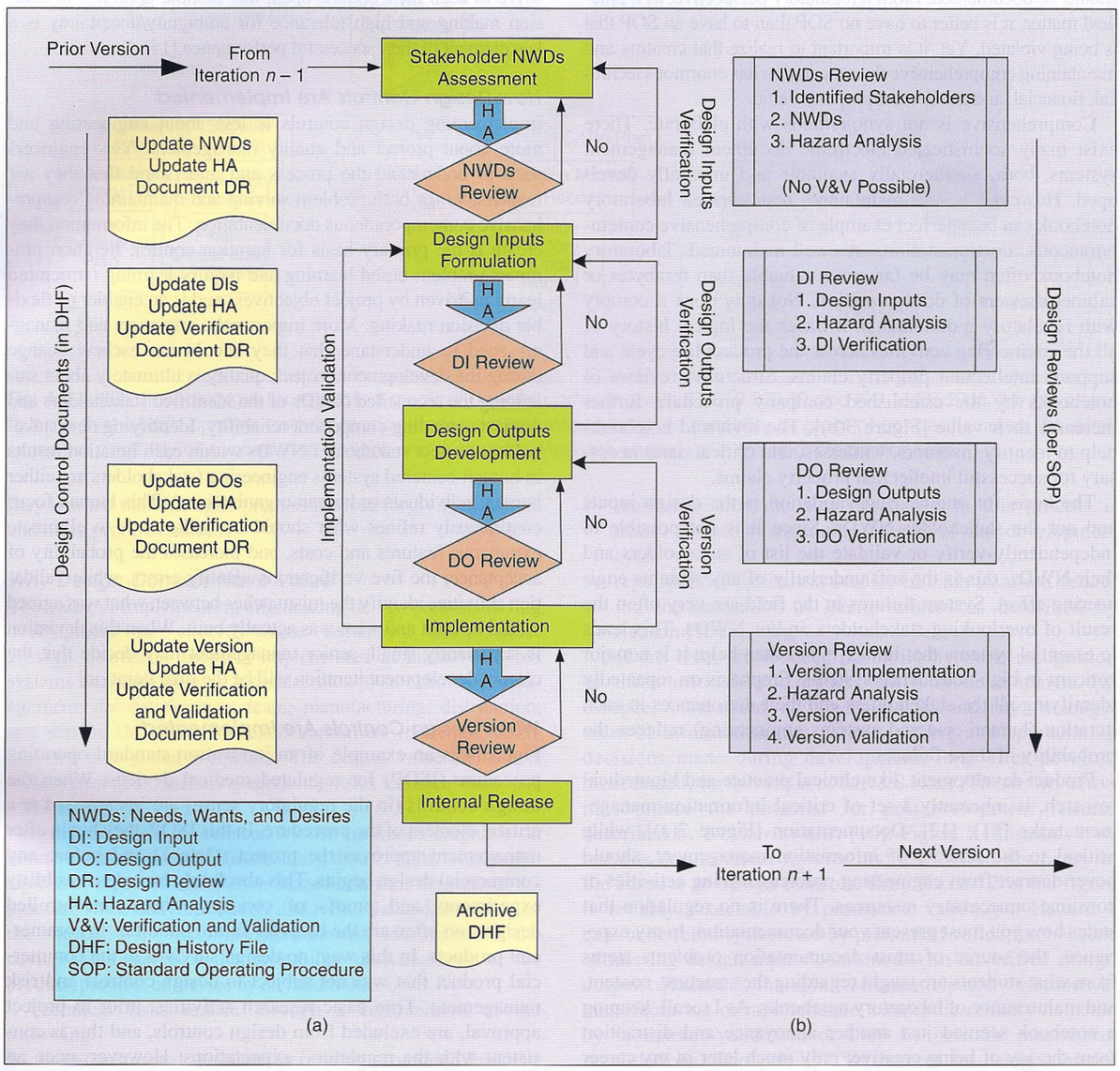


Fig. 3. The flowchart depicts some arbitrary iteration of the design control process and identifies (a) requisite documentation activities and (b) design review targets. Three of the five verification activities are identified; the two risk management verification activities (ISO 14971 §6.3) are omitted for clarity.

stakeholders improves with each iteration, if there are multiple iterations and real opportunities for the development team to learn [10].

Comprehensive Documentation

Standard operating procedures (SOPs) and their requisite documentation work products have become the bane of engineers' existence. Intended to be a powerful management technique that standardizes the process, character, and quality of work, it has instead become an annoyance to engineers, a frustration for management, and a regulatory vulnerability. From the engineers' perspective, SOPs are fine guidelines, but everyone knows the work changes, and no one is keeping up with the SOPs. From managements' perspective, SOPs make excellent training materials in addition to establishing the manner in which the work (e.g., design controls) should be accomplished and the achievements should be documented. From a regulatory perspective, as a practical matter, it is better to have no SOP than to have an SOP that is being violated. Yet, it is important to realize that creating and maintaining comprehensive documentation has enormous technical, financial, and intellectual property value.

Comprehensive is not synonymous with elaborate. There exist many sophisticated electronic document management systems, both commercially available and internally developed. However, a well-maintained, hand-written laboratory notebook can be a perfect example of comprehensive contemporaneous documentation. A well-maintained laboratory notebook often may be far more valuable than terabytes or cabinet drawers of documentation. Not only does it comply with regulatory requirements, it traces the logical history of all the engineering activities across the product life cycle and supports intellectual property claims. Structured reviews of notebooks by the established company procedure further increases their value [Figure 3(b)]. The reviewed notebooks help to identify inventors, witnesses, and critical dates necessary for successful intellectual property claims.

The basis for engineering validation is the design inputs and not the stakeholder NWDs. Since it is not possible to independently verify or validate the list of stakeholders and their NWDs, this is the soft underbelly of any systems engineering effort. System failures in the field are very often the result of overlooking stakeholders and/or NWDs. This leads to essential systems that hinder rather than help; it is a major concern in high-confidence systems. Emphasis on repeatedly identifying all the stakeholders and their dissonances in each iteration (human-centered systems engineering) reduces the probability of these failures.

Product development, like clinical practice and biomedical research, is inherently a set of critical information-management tasks [11], [12]. Documentation [Figure 3(a)], while critical to the process of information management, should never distract from engineering problem-solving activities or consume unnecessary resources. There is no regulation that states how you must present your documentation. In my experience, the source of most documentation problems stems from what students are taught regarding the structure, content, and maintenance of laboratory notebooks. As I recall, keeping a notebook seemed just another annoyance and distraction from the joy of being creative; only much later in my career did I appreciate the critical value of capturing information and experience in my notebook, especially when I needed that information after having forgotten it.

Flexible Decision Making

Too much time spent on any particular step in a process during a single iteration is usually counterproductive. The optimal design invariably changes with each iteration. Trying to finalize any subsystem usually constrains future decision-making options, resulting in suboptimal results. This is initially anathema to most engineers and managers, who wish to believe that a known percentage of the project is complete. Once they experience this as an agile process, the majority become far less skeptical and insecure about not constantly computing percentage complete. (All agile processes are iterative, but not all iterative processes are agile. A key attribute of an agile process is that each iteration is time constrained.) Instead, they focus on estimating the remaining risk with each ensuing iteration [13], fully aware that in this process the resource costs of change are no longer onerous. It has been my observation that high-performing individuals usually strive to keep their options open; this flexible approach to decision making and high tolerance for ambiguity/uncertainty is a key element of their successful performance [14].

How Design Controls Are Implemented

Implementing design controls is less about engineering and more about project and quality management. Yes, engineers need to understand the process and understand that they are responsible for both problem solving and maintaining comprehensive contemporaneous documentation. The information they create is the primary basis for iteration control. Iteration promotes problem-based learning and inquiry learning (structured learning driven by project objectives) and is an enabler of flexible decision making. More importantly, engineers and managers need to understand that they should not eschew change during the development project; quality is ultimately about satisficing the reconciled NWDs of the identified stakeholders and not just extending component reliability. Identifying new stakeholders or new stakeholder NWDs within each iteration results in human-centered systems engineering (stakeholders are either human individuals or human organizations). This human focus continuously refines what should be built, tends to eliminate extraneous features and costs, and increases the probability of acceptance; the five verifications identify errors, while validation activities identify the mismatches between what was agreed would be built and what was actually built. When this deviation is sufficiently small, senior management may decide that the current development iteration will be the final iteration.

When Design Controls Are Implemented

Figure 4 is an example of an innovation standard operating procedure (ISOP) for regulated medical devices. When the design controls (in the regulatory sense) are instantiated is a critical element of the procedure. In this ISOP, they begin after management approves the project (Gate 1) and before any commercial design begins. This absolutely includes feasibility experiments and proofs of concept, whose uncontrolled designs too often are the basis for future liabilities in commercial products. In this way, no design survives in the commercial product that was not subject to design controls and risk management. True basic research activities, prior to project approval, are excluded from design controls, and this is consistent with the regulators' expectations. However, once an organization has implemented design controls in one or two projects, I have found that the members of the project teams will often just do it as a matter of practice, probably because

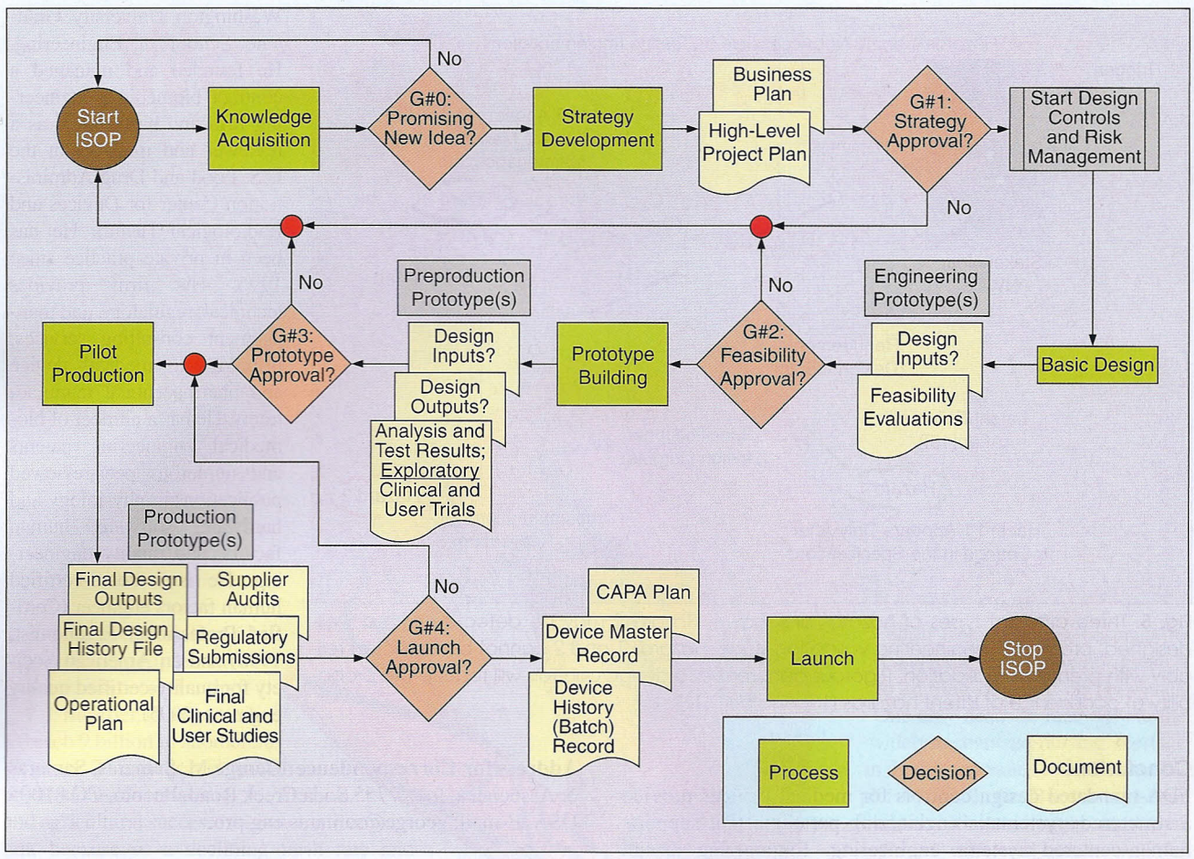


Fig. 4. An ISOP flow diagram for regulated medical devices complying with 21 CFR 820.30 and ISO 13485:2003. It is important to note that design controls and risk management are instantiated prior to feasibility or proof-of-concept studies. This eliminates the possibility of uncontrolled designs and their associated risks ending up in the final production units.

they learned that this agile process is more forgiving of errors and more prone to success.

Why Design Controls Have Value

The value of implementing engineering design controls during development far exceeds mere regulatory compliance. Design controls, properly implemented as human-centered systems engineering, have tangible value to stockholders; management; the development team; manufacturing, distribution, and service (MD&S) personnel; consumers (both purchasers and end users); and regulators. The central value element is reduction of risks, including economic risks, technical risks, and MD&S risks.

Economic risks are reduced by the enormous emphasis on iterative reassessment of stakeholders, their NWDs, and the discipline of repeated cumulative hazard analyses throughout the process. These activities clarify and refine the understanding of the intended uses, users, use environment, and lifetime of the device. This increases the probability of acceptance of the product, process, or service by all stakeholders.

Technical risks reduce through a combination of the following:

- time-constrained iterations permitting structured learning
- comprehensive contemporaneous documentation providing efficient traceability and supporting flexible decision making
- realization that the cost of change in this process is nearly flat from beginning to end of the development cycle
- repetition of validation studies in each iteration reducing the incidence of latent failures as described by Reason [15].

MD&S risks are reduced primarily by overt recognition of these critical stakeholders early in development (in essence, concurrent engineering). Secondarily, management of information that permits tracing and appreciating decisions made during development, long after development ended, further reduces MD&S risks. These two activities reduce the potential for postdevelopment failures described by Dekker as drift [16]. Figure 5 illustrates how hazards due to latent failures and drift avoid discovery in the absence of design controls. Only Hazard 2 in the figure is reliably detectable by premarket validation. Hazard 1 cannot be reliably detected by premarket validation because of missing design inputs combined with unverified design outputs. Hazard 3, the result of unanticipated variations in manufacturing or maintenance, cannot be reliably detected by premarket validation or even by periodic postmarket revalidations. The principle of correct by design, fundamental to the development of high-confidence systems, presupposes effective design controls throughout the full life cycle (from lust to dust).

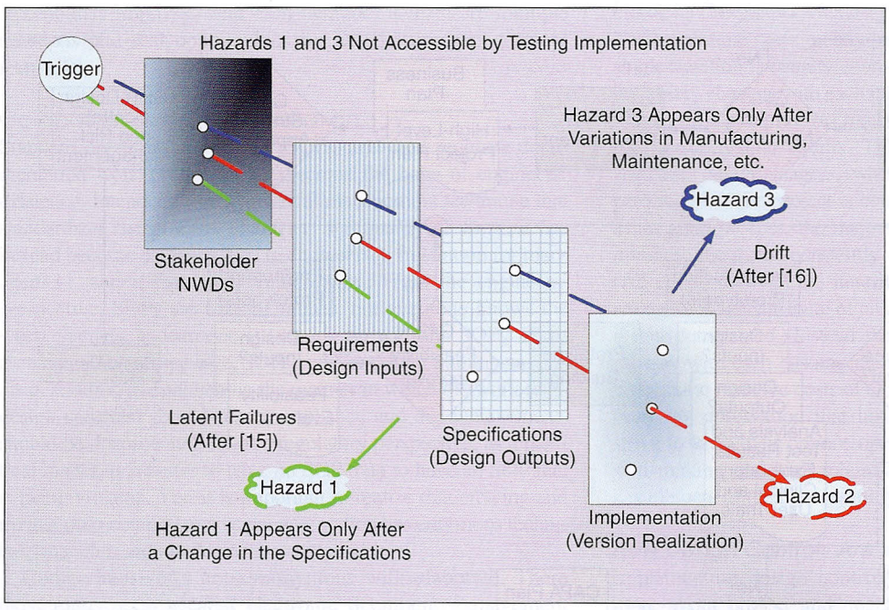


Fig. 5. Three different types of hazards are shown. Hazard 2 may be detected with a well-designed, premarket engineering validation study. Hazards 1 and 2 cannot be detected reliably with premarket validation. Rigorous adherence to design controls will reduce the probability of occurrence of latent hazards (15) and drift (16).

Conclusions

FDA-mandated design controls for medical devices provide a structured, systematic engineering paradigm that supports human-centered systems engineering. Engineering design controls are nothing more than the fundamental elements of classical systems engineering. The human focus is enabled through the iterative reidentification of stakeholders, reassessment of their NWDs, and reconciliation of their evolving/conflicting NWDs. The implementation of design controls and embedded risk management must begin prior to commercial development to reap the full benefit of the approach; partial approaches dilute or negate the effectiveness and efficiency of this nearly century-old systems engineering paradigm. Properly employing engineering design controls is a strategic business decision. The central value of this proposition is the reduction of economic, technical, and operational risks for both producers and consumers; regulatory compliance is a secondary benefit. Misuse or abuse of design controls only undermines long-term profitability and increases the risks to the consumer.

George M. Samaras received his B.S.E.E. degree in electrical engineering in 1972 and M.S. and Ph.D. degrees in physiology from the University of Maryland in 1974 and 1976, respectively. He received a D.Sc. degree in 1992 from the George Washington University, Department of Engineering Management and Systems Engineering. He has been a professional engineer, licensed in Maryland since 1980. He worked as a professor at the University of Maryland School of Medicine and George Washington University Graduate School of Engineering. He founded and managed a contract biomedical engineering firm and has worked as a reviewer and manager at the U.S. Food and Drug Administration Center for Devices and Radiological Health. He has been in private practice since 1996. 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 board-certified human factors engineer (Certified Professional Ergonomist, 1998) and an American Society for quality-certified quality engineer (2005).

Address for Correspondence: George M. Samaras, Samaras & Associates, Inc., 7755 Soda Creek Road, Pueblo, CO 81005 USA. E-mail: george@samaras.eng.pro.

References

[1] G. M. Samaras and R. L. Horst, "A systems engineering perspective on the human-centered design of health information systems," *J Biomed. Inform.*, vol. 38, no. 1, pp. 61–74, 2005.
[2] United States Food and Drug Administration Code of Federal Regulations, Part 820 (Quality System Regulation), Final Rule, Oct. 1996.
[3] *Medical Devices—Quality Management Systems—Requirements for Regulatory Purposes*. 2nd ed., International Standard ISO 13485:2003(E).
[4] H. A. Simon, *Models of Man: Social and Rational*. New York: Wiley, 1957.
[5] G. M. Samaras, "Human-centered systems engineering: Building products, processes, and services," in *Proc. 2010 SHS/ASQ Joint Conf.*, Atlanta, GA (CD-ROM), Feb. 26, 2010.
[6] W. W. Royce. (2008, Oct. 24.). Managing the development of large software systems, *Proc. IEEE WESCON*, Aug. 1970, pp. 1–9 [Online]. Available: http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf
[7] D. Parnas and P. C. Clements. (2008, Oct. 24). A rational design process: How and why to fake it. *IEEE Trans Softw. Eng.*, vol. 12, no. 2, pp. 251–257. [Online]. Available: http://users.ece.utexas.edu/~perry/education/SE-Intro/fakeit.pdf
[8] *Medical Devices—Application of Risk Management to Medical Devices*. 2nd ed., International Standard ISO 14971:2007(E).
[9] G. M. Samaras, "An approach to human factors validation," *J Valid. Technol.*, vol. 12, no. 3, pp. 190–201, 2006.
[10] J. G. March, "Exploitation and exploration in organizational learning," *Organization. Sci.*, vol. 2, no. 1, pp. 71–87, 1991.
[11] V. L. Patel, E. H. Shortliffe, M. Stefanelli, P. Szolovits, M. R. Berthold, R. Bellazzi, and A. Abu-Hanna, "The coming of age of artificial intelligence in medicine," *Artif. Intell. Med.*, vol. 46, no. 1, pp. 5–17, 2008.
[12] J. H. Poore, "A tale of three disciplines ... and a revolution," *IEEE Computer*, vol. 37, no. 1, pp. 30–36, Jan. 2004.
[13] C. Larman, *Agile and Iterative Development: A Manager's Guide*. Boston: Addison-Wesley, 2004.
[14] P. C. Nutt, "Flexible decision styles and the choices of top executives," *J. Manage. Stud.*, vol. 30, no. 5, pp. 695–721, 1993.
[15] J. Reason, *Human Error*. Cambridge: Cambridge Univ. Press, 1990.
[16] S. W. A. Dekker, *Ten Questions About Human Error: A New View of Human Factors and System Safety*. NJ: Lawrence Erlbaum Associates, 2005.