FDA AND THE REGULATION OF MEDICAL SOFTWARE

Paul T.H. Kim, Office of Policy, Food and Drug Administration

This paper traces the development of the Food and Drug Administration's (FDA) regulation of computer software, as discrete products with medical applications as well as components of regulated medical devices. The earliest Agency deliberations of software are summarized, as are the Agency's broad policy priorities and the concerns of developers, medical industries, and user communities over the potential scope and consequences of FDA regulation.

For the purposes of discussion, the medical applications of software can be divided into three general types. First, software may control, regulate, or operate in tandem with medical devices. This type of software can be described as **medical device software**. Second, software may be used in the manufacture, assembly or testing of medical products including devices and pharmaceuticals. This type of software may be labeled **process software**. Finally, software may provide a medical function independent of its hardware system. These kinds of software can be described collectively as **medical software**.

FDA claims broad authority to regulate all of these types of software. FDA regulates medical device software as "component[s], part[s], or accessor[ies]" of medical devices. Process software are subject to regulation under FDA's Good Manufacturing Practice (GMP) requirements for medical devices (21 CFR 820). Medical software is regulated according to FDA's draft policy for the regulation of computer products (1987, revised 1989). This policy delineates FDA's scope of authority over hardware and software alike, and exempts certain categories of software from regulation.

There are two outstanding software-related issues which require further FDA policy development. First, FDA must continue its current efforts to improve the reliability of medical device software. It is well known that improving design practices is the most effective means of preventing software-related problems. FDA's Center for Devices and Radiological Health (CDRH) must expand its cooperative efforts with industry to further educate device manufacturers on current regulatory requirements. Second, FDA should finalize the 1989 draft policy for computer products. The current draft relies upon concepts such as "competent human intervention" which require further refinement. Although device manufacturers and software vendors are generally aware of the policy, there is continuing uncertainty regarding its application.

1. MEDICAL DEVICE SOFTWARE: DESIGN AND QUALITY ASSURANCE

1.1 Why software is "different"

Ensuring the quality and reliability of medical device software requires FDA to approach it somewhat differently than hardware. Although software frequently acts as a functional equivalent to a physical component such as an analog display or a mechanical fixture, it also embodies properties which set it apart from hardware. Viewed as a series of interconnected logic processes, software "can exhibit discontinuities with jumps and branches of such complexity that repeatability is difficult or impossible to prove."[1] To some extent, software development is also an art as well as a science. A piece of software can embody a developer's individualistic design choices which increase its opacity to external analysis.[2]

[1] R. Murfitt, "U.S. Government regulation of medical device software: a review," *Journal of Med Eng & Tech*, 14:3 (1990), 111.

[2] V. Brannigan, "Software quality regulation under the Safe Medical Devices Act of 1990: hospitals are now the canaries in the software mine," *Proceedings of the 15th Annual AMIA Symposium on Computer Applications in Medical Care*, November, 1991.

Most importantly, software which performs sophisticated tasks can be very complex. For example, "radiation treatment therapy planning systems" are powerful software packages that plan a cancer patient's radiation therapy protocol (e.g. calculating dosage and treatment volume, delineating critical structures and tumor contours)-- after which computer delivery and verification systems ensure that the radiation is delivered properly. As another example, implantable cardioverter-defibrillators will be marketed in the near future. These products may include as little as a few kilobytes of preset software and less than a megabyte in an external interrogator. This relatively small amount of software, however, can allow for almost 10^{45} different programmable settings.

1.2 FDA's regulatory strategy

To cope with the difficulties of assessing software quality, CDRH has placed a strong emphasis upon the importance of a well-documented, "rational" software development process and robust quality assurance (QA) practices. As a result of the Center's activities, FDA is widely regarded as "a principal advocate for the improvement of software quality within the medical device industry."[3] The importance of well-executed design and QA to software cannot be overstated. The vast majority of software-related device failures are attributable to design-related errors. Defects in initial software design are also much easier in principle to detect than in actual code--and the earlier that defects are detected, the cheaper they are to correct.

Consequently, CDRH attempts to capture software quality in products undergoing 510(k) and PMA review by examining manufacturers' "preproduction quality assurance" in the software design and development phases. The manufacturer is responsible for mitigating risk by seeking problems that may be safety hazards in the software rather than reacting to the manifestation of problems during its operation.

1.3 FDA's reviewer guidance

The critical document used by CDRH in 510(k) and PMA reviews of medical device software is the "Reviewer Guidance for Computer Controlled Medical Devices". The reviewer guidance was issued in draft form in 1988, and prompted a joint HIMA/NEMA rewrite from the medical device industry in 1989. The final draft was issued by CDRH in 1991, setting out the Center's general policy on pre-market review of medical device software.

Although officially intended for use with 510(k) reviews, the guidance is relevant to PMA submissions as well. Briefly, the guidance establishes what data is expected in a computer-controlled device submission and directs reviewers to categorize products in one of three "levels of concern".[4] The level of concern reflects the level of regulatory control and is dictated by the device's intended use, the role of software in the device, and the potential risks to patients. The higher the level of concern, the more extensive the documented QA and development activities expected from manufacturers.

The reviewer guidance has been the focus of FDA and industry attention on software issues. Since 1987, however, the discussion has shifted notably: industry presentations at the regular HIMA conferences on software have moved away from addressing "Should this be done?" to "How to satisfy the regs."[5] As an ODE reviewer commented, "The industry has come a long way."[6] On the other hand, there is considerable concern among manufacturers regarding the implementation of the reviewer guidance. One industry consultant commented that the device industry is "very confused" about how the reviewer guidance should be used to guide product

[3] Murfitt, 111.

[4] A "major level of concern" would apply to a device or software failure resulting in death or serious injury, a "moderate level" to minor to moderate injury or if the device affects the patient only indirectly, and a "minor level" for no expected injury to the patient.

[5] Interview, CDRH, September 24, 1992.

[6] Interview, CDRH, January 11, 1993.

development.

2. PROCESS & MEDICAL DEVICE SOFTWARE: IMPACT OF REVISED DEVICE GMPs

2.1 Postmarket controls through device GMPs

CDRH uses its authority under the current device GMPs (21 CFR 820) to ensure that device manufacturers and software vendors support adequate levels of QA and documentation for their products, updates and revisions. The current GMPs require manufacturers to establish QA programs, staff them with trained personnel, document their activities, and assure the acceptability of device components and labelling. According to a CDRH guidance document, a QA program should include "procedures for assuring approval or rejection of contract-supplied software for...medical devices, control of manufaturing processes, and use in quality assurance activities."[7]

The proposed revisions to medical device GMPs represent significant changes to FDA policy towards both medical device and process software. Currently under review at FDA's General Counsel, the proposed revisions are due to be published in spring 1993. In addition to ensuring regulatory compatibility with the ISO 9001 standards being adopted by the European Community (EC), the revised GMPs will allow FDA to more effectively regulate marketed medical device software and process software. In the event of changes to medical device software, for example, inspectors had previously had access to the Device Master Record (DMR) which included the device specifications.

A change being considered under the revised GMPs would allow inspection of software design and validation documentation as well. These records would enable determination of whether the development process was orderly and controlled, ie. whether vendors have maintained a satisfactory level of control over the updating and maintainance of their software. CDRH rightfully expects this new capability to prove extremely important to its ability to ensure the quality of medical device and process software. As one CDRH manager put it, "Design is the whole ballgame with software and effective [design] controls will become routine through GMP inspections and the works."

3. BLOODBANK SOFTWARE

3.1 Critical reliance on software

The issue of bloodbank software arose in 1988-89, when a software-related failure in an automated optical testing unit led to the false validation of "millions" of plasma units at a plasmapheresis plant. An altered interface between the testing unit and a mainframe computer improperly altered negative (fail) values to absolute (pass) values. The defective bloodbank software system was used widely at the time by bloodbanks to prevent the release of unsuitable units identified on the basis of five or six quality tests. The software contained serious design defects. Units were evaluated using data from only two or three of the available tests. Additionally, the software had not been validated to prevent dual user access, resulting in other data problems.

During the GMP investigation, these and other problems were identified. Among the most important concerned "change control". Over several years, the software vendor had implemented several hundred changes to its software but did so site-specifically. In other words, if a change was made at one location, the vendor did not inform its other users of the change. Sometimes the corrections were done remotely via modem, and the users at the correction site were never informed.

This situation raised the issue of software to the Commissioner's attention in 1989. It was a timely example of a system-critical software package requiring FDA regulatory action. As a result, bloodbank software was removed from the 1987 draft policy list of products exempted from regulation. The driving factor in this decision was the fact that, unlike drug manufacturing

[7] *Computerized devices/processes guidance: application of the medical deivce GMP to computerized devices and manufacturing processes*, Office of Compliance and Surveillance, CDRH, FDA, May 1992.

facilities, the bloodbanks relied critically upon these systems. Center for Biologics Evaluation and Research (CBER) decided that the level of concern justified regulating bloodbank software vendors as well as the bloodbanks themselves. Guidance was issued by CBER in the form of two memoranda (1988, 1989) to bloodbanks, instructing them to validate their systems and software, and to rely upon known vendors. In 1990, HIMA also sponsored a symposium on the topic of bloodbank software.

3.2 CBER management plan

Currently, CBER continues to regulate bloodbank software vendors. CBER is in the process of implementing a comprehensive management plan for bloodbank software, which ranges from expanding its in-house knowledge of existing vendors through to surveillance, inspections and enforcement. CBER's authority is already generally well recognized by bloodbanks and software vendors. These stakeholders are cooperating, particularly through industry associations (HIMA, AABB), with CBER on defining best practices for system validation.

4. FDA DRAFT POLICY FOR THE REGULATION OF COMPUTER PRODUCTS

4.1 Historical development of the policy

FDA first established a task force to investigate software and artificial intelligence in 1981, under the leadership of Dr. Carl Bruch. Their investigations stemmed in large part from discussions at HIMA seminars as well as with academics and manufacturers. In 1987, FDA issued the draft policy and incorporated comments into its 1989 revision. In large part, the policy was drafted to reflect the regulatory *status quo*, with the exemption for the "textbook function" of libraries and the future exemption for expert systems included to placate the National Library of Medicine (NLM). One CDRH manager who contributed to the policy characterized it as "a reflection of prevailing practices...it served as a pressure valve" which retained FDA's authority over a wide range of products by classifying them all --from "general purpose" products to expert systems-- as devices but generally leaving them alone.[8]

In his September 1986 speech on the NLM's 150th anniversary, Commissioner Frank Young emphasized the FDA's desire to maintain the lowest level of control needed to provide a reasonable assurance of safety and effectiveness in regulated computer products. In addition to explicitly folding "library" functions outside of FDA's jurisdiction, Young explained that FDA sought "guidance as we approach the field of artificial intelligence.

4.2 Policy basics and next steps

The draft policy sets out the following concepts: 1) traditional "library", general acounting and communication, and education software or computer systems are outside of FDA jurisdiction; 2) all other software or computer systems which meet the definition of medical device are within FDA jurisdiction; and 3) FDA exempts several categories of software or computer systems from active regulation.

Since most medical devices had already been classified under the 1976 amendments, only "unclassified" hardware and software marketed for medical use were meant to be covered by the draft policy.[9] Three exemptions are provided for software or systems intended for "general purpose[s]" (such as PCs or database software), for teaching or non-clinical research, and for personal use by professionals who have altered or customized them.

The policy also defines those classes of products which require FDA notification prior to marketing and those requiring premarket approval. Finally, a future exemption is provided for products such as decision-support or expert systems "intended to involve competent human intervention before any impact on human health occurs".

[8] Interview, CDRH, September 24, 1992.

[9] Charles S. Furfine, "The FDA's policy on the regulation of computerized medical devices," *M.D. Computing*, 9:2 (1992), 97.

The draft policy requires finalization for two important reasons. First, although the policy is well known (if not fully understood) by its stakeholders, its "draft" status perpetuates concern regarding future regulatory action. "There is tremendous uncertainty," according to one industry consultant. "System vendors will remain in limbo for the foreseeable future." Or as a device manufacturer insisted:

> "Manufacturers don't know what to plan for their systems five to ten years down the road, nor what to comply with...The economic impact of these policies will force smaller vendors out of the market, resulting in less competition and limited product selection."[10]

Second, some of the key terms must be further refined to end any ambiguity regarding the policy's application. "Competent human intervention" and "opportunity" for such intervention are key terms which CDRH managers and manufacturers alike recognize as needing clarification. For example, despite the inapplicability of the policy to their products, bloodbank software vendors have claimed that their systems incorporate "competent human intervention". However, CDRH inspections found that system operators soon relied upon their computers uncritically; a natural human tendency to suspend active judgement and to trust reliable but not infallible systems.

Additionally, there have been some difficulties in implementing the policy uniformly. Some submissions have been received for exempted devices, while the argument has been made that regulatory exemptions on the basis of "competent human intervention" could logically be extended to other devices. For example, could home test kits or similar products be regarded as providing for "competent human intervention"?

The concept of "competent human intervention" rests heavily upon several definitions which have been inadequately articulated to the policy's stakeholders. Competence, for example, rests upon an "understand[ing] of the clinical implications of the device and its output" as defined in the draft Q&A-- a document which for various reasons was never issued publicly. The fundamental difficulty with the concept of "competence" is how it depends heavily upon whether a system and its workings are comprehensible or whether they represent an inaccessible "black box" to the user.

4.3 Black boxes and expert systems

For example, it is well understood that a system's capacity to explain its reasoning is an important criterion for defining it as an expert system. Expert systems provide advice to a physician in emulation of a human expert. Although expert systems generally do better than the average M.D., physicians have long considered a system's ability to explain its advice as its most important feature. This capacity is essential to increasing the willingness of clinicians to accept the advice offered, to persuading them that unexpected advice is appropriate, and to helping them reject advice when it is incorrect or inappropriate.

Although increasing numbers of expert systems are being developed for use in clinical specialties, there are no indications that they pose an active risk to human health. Of the handful of software products reviewed in *JAMA* from 1990-1993, none would fall under the policy exemption for expert systems. However, expert systems have become far more accessible to a broader audience. Commercial packages now provide expert system "shells" which can be tailored to specific needs or custom requirements. It is far from clear, however, whether expert shells retain an adequate explanatory capacity following customization, e.g. for a technical or medical appplication.

5. SOFTWARE PRACTICES IN OTHER FEDERAL AGENCIES

5.1 Summary of agency software policies review

A look beyond the FDA and the medical device industry provides insight into the way that quality and reliability are achieved by other regulators and consumers of computer software.

[10] "All talk and no action: information system users face regulatory challenges," *Computers in Healthcare*, August 1992, 31.

Other Federal agencies have extensive track records in procuring or regulating software and software-dependent systems. How do these other agencies assure the quality of software they procure or regulate?

The four Federal agencies[11] reviewed for this paper are currently improving or expanding their programs for software quality assurance. Of these agencies, FDA's responsibilities are closest to those of the Nuclear Regulatory Commission (NRC). Neither agency is directly involved in the acquisition or development of software; both agencies review the products developed and QA conducted by a regulated industry; and both agencies are still moving along a learning curve of how to regulate software development effectively. The striking difference lies in the diversity of products that FDA is obliged to review and approve; NRC is dealing with the narrower issue of automating reactor instrumentation and controls.

Parallels may also be drawn from the experiences of the other Federal agencies. A few --most notably DoD and NASA-- have already made significant changes to their software practices as a result of external scrutiny and criticism. Sporadic problems in regulated software, such as FAA-approved flight systems, have also generated media attention and Congressional inquiries.

There has been progress towards improving interagency communications on software, particularly through the COMPASS program organized by Department of Commerce's National Institute for Standards and Technology (NIST). In addition, several agencies have been actively collaborating with the Institute of Electrical and Electronics Engineers' (IEEE) efforts at codifying standards for software development. It is also significant that three of the four surveyed agencies have sought advice and technical assistance from the Software Engineering Institute (SEI), a highly regarded government-funded software research center.

While DoD has been SEI's primary client since the institute's inception, both NRC and FAA recently entered discussions with SEI. Both agencies, an SEI engineer noted, approach software safety with a unachievable "zero tolerance" for failures. As he pointed out, not even the best experts can provide silver bullets for their clients' problems-- "software quality is a product of "multiple capabilities, practices, and influences."

The views expressed in this article are solely those of the author and do not represent the views of the Food and Drug Administration, Department of Health and Human Services, or the United States.

[11] Department of Defense (DoD), Federal Aviation Administration (FAA), National Aeronautics and Space Administration (NASA), Nuclear Regulatory Commission (NRC).