Published on *Medical Electronics Design* (http://www.medicalelectronicsdesign.com)

Medical Device Mechatronics Maturity

G.M. Samaras
Created *2012-05-23 13:44*
[1]

Medical Device Mechatronics Maturity [2]

May 23, 2012
By: G.M. Samaras

0
Share [3]

Find more content on: Design [4] Feature [5] Software [6] Technology [7]

Even if your only motive is maximizing profit, not maximizing your organization's mechatronics maturity is a waste of your resources.

Mechatronic medical devices are an important historical innovation. Hardware, combined with information-driven software processes, offers increasingly sophisticated functionality, but it also creates important problems with quality. Integration of the engineering work of various hardware, software, and human factors disciplines is a major challenge. Validation and verification testing are about avoiding mistakes in engineering development, deployment, and maintenance. Human factors practitioners call these mistakes errors. There are two fundamental rules regarding errors:

- Rule 1—All errors are made by humans
- Rule 2—All errors are experienced by users.

One categorization of errors of critical importance to engineers is: system use versus individual user errors. Mechatronics maturity is a measure of how good you are at avoiding system use errors. Testing is necessary, but never sufficient. The specific processes embedded in your organization help us estimate mechatronics maturity ... and that is a predictor of new product development (NPD) quality, which is one reason why FDA believes that quality system management audits have value. From a business perspective, a low maturity NPD process is more costly and less time efficient; a high maturity NPD process is very cost and time efficient. Even if your only motive is maximizing profit, not maximizing your organization's mechatronics maturity is a waste of your resources.

Introduction

An important historical innovation [8] in healthcare has been the advent of mechatronic [9] medical devices—devices that integrate sensor and effector mechanical and electrical hardware (HW) with information-driven software (SW) processes into a potentially synergistic whole, offering increasingly sophisticated functionality. They range the gamut from simple positioning systems, to infusion pumps, to robotic surgical devices and healthcare information technologies.

While the use of mechatronics offers enormous potential for increasingly sophisticated functionality, it also presents equally large quality problems with interdisciplinary development, deployment, utilization, and maintenance.[4] These are not merely technical issues (e.g., promoting integration of HW and SW development by automatically generating a new hardware abstraction layer [10] with each HW revision), but also organizational issues (e.g., preventing development from occurring in independent silos) and project management issues (e.g., emphasizing and prioritizing quality milestones over schedule and budget milestones).[2, 1] How an organization deals with the HW, SW, and all-encompassing human factors (HF) issues may be viewed as a measure of mechatronics maturity.

Why HF issues? In the final analysis, all products, processes, and services exist solely because they are perceived to have utility or esthetic value for some humans. All engineered systems require humans. Yes, even isolated, automated, unsupervised systems require humans; they are called installers and maintenance personnel. And, what we engineers think of as the "system" is really not. The functioning system is the combination of our product, process, or service plus all the humans (operators, maintainers, etc.) and their organizations; without all these humans and their organizations, our "system" cannot function and can have no value.

Quality exists only in the eyes of the human stakeholders [11] and their organizations. Quality may be defined as: the degree to which the system satisfices [12] the needs, wants, and desires (NWDs) of ALL the stakeholders, which means the degree to which you obtain a good result that is good enough, though not necessarily the best, for each of all the stakeholders.[10] From this simple definition, all other measures of quality can be derived. The four top-level NWDs of all stakeholders are: safety, effectiveness, efficiency, and stakeholder satisfaction (SEES).[12] For medical devices, this means products that are safe and effective, efficient to produce and use, and satisfying to all stakeholders, including both individuals and their organizations.

Mechatronic Medical Devices

Mechatronic devices are the result of integration of the engineering work of various HW disciplines (mechanical, electrical, etc.), various SW disciplines (embedded, application, etc.), and—of necessity, as many are now beginning to realize—various HF disciplines (micro-, meso-, macro-, and mega-ergonomics).[12] Practitioners of these various disciplines speak different technical languages and have very different perspectives of the engineering process; in many cases, some of the SW and HF

practitioners have no engineering background. This creates significant engineering management problems that are typically not resolved with standard project management methods. We often pretend that these are problems with the HW, SW, and HF practitioners; in fact, they are engineering management [13] problems. Permitting HW, SW, and HF work to proceed un-integrated in individual "silos" is a regressive engineering management approach that thwarts efficient communication & coordination, thwarts the effective implementation of design controls, and invariably reduces quality of the developed product, process, or service—stifling innovation.[2, 11]

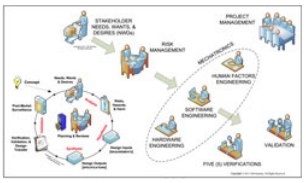
Design controls are nothing more that the fundamental principles of classical systems [14] engineering, a well-established branch of engineering in existence since the 1940s.[8] It is important to remember that design controls are an engineering process; yes, they produce documentation, but that documentation can be sticky notes (e.g., Scrum [15], Kanban [16]), hand-written notebooks, or outputs from a word processor. The form of the documentation should not drive the engineering process. The documentation exists solely to demonstrate what process was actually followed and what decisions were actually made during that process. Forget about FDA audits; an important aspect of design controls is about preserving and protecting intellectual property and institutional knowledge.

Figure 1 illustrates the mechatronic development process in the medical device design control scheme. In the lower left corner is a simplified diagram of the scheme, emphasizing the cyclical nature of development, deployment, and maintenance.[8, 11] Why cyclical? Because engineering development is about learning (knowledge generation) and we do not learn in a linear "waterfall"; human learning is always non-linear, incremental, and builds upon experience and repetition, usually gained from testing – some of which is formal, but much of which is informal, accidental, and experiential.[11]

On the right of Figure 1 is a linearized, idealized view of selected parts of the iterative process emphasizing the interdisciplinary character of mechatronics. It begins with identification of the stakeholders and their NWDs and is followed by the beginning of risk management [17]. A subset of the discovered stakeholder NWDs, which is deemed currently technologically and economically feasible, usually becomes the initial set of design inputs. Design inputs are a natural language, operationalized [18] statement of the engineering problem (the engineering requirements [19]).[11] This engineering problem is posed to the HW, SW, and HF practitioners; their work product is the set of design outputs. Design outputs are the engineering specifications that tell manufacturing exactly how to make and test the medical device. I have intentionally omitted many other elements of the process, including the very important concurrent engineering [20] elements, so as to draw your attention to two processes that cause a lot of confusion: engineering verifications and validation.

The terms verification and validation have been extensively misused, abused, interchanged, and their definition can even vary by industrial sector. In the medical device design control scheme, validation means that you developed the right system (the engineers correctly solved the problem captured by the design input process); verification means that you developed the system the right way.[5, 9, 11] For example, if I ask you to build a non-contact blood pressure monitor and, instead, you build a non- contact brainwave (EEG) monitor, you may have done everything the right way from an engineering perspective, but you did not develop the right system. And, while I and many others would love to have a non-contact EEG monitor, you would still fail validation.

Figure 2 shows that, while there can only be one type of design validation, there are five different types of design verifications. Three types of verification are from classical systems engineering (verification of requirements, specifications, and implementation).[8, 11] The other two types of verification are from risk management; they are verification that you properly applied your proposed mitigation and verification that your properly applied mitigation actually reduced the target risk.[6] Unlike validation, verifications are internal error-correcting mechanisms in the engineering process that reduce the probability of mistakes during the development, deployment, or maintenance effort.



[21]

Figure 1. [21] Mechatronics in the design control scheme. Left-side insert adapted from [11].

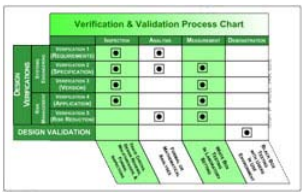
Validation lets you determine whether you developed what your stakeholders asked you to develop (in the design inputs)—did you develop the right system? Validation can occur at the system level, subsystem level (e.g., software validation), or module level (e.g., validation of third-party modules designed for interoperability). System validation is the final defense against Type III errors [22]: "correctly solving the wrong problem." That is our brainwave monitor example, although in most instances, the difference is not quite so radical—and often may be quite subtle and difficult to detect during development. System validation should occur once at the end of each iteration (sprint) after the new version of the system is built; it should prevent you from proceeding to the next iteration (sprint), if you are not developing the right system.

Unfortunately, as we shall discuss in the next section, validation is not foolproof.[9] Nevertheless, confusion or incorrect use of verifications and validation will invariably result in low quality—in addition to noncompliance with quality system regulations and consensus standards—and, ultimately, stifle innovation.

Eяrors, Erяors, Errors

Validation and the five verifications are all about avoiding mistakes in engineering development, deployment, and maintenance. HF practitioners call these mistakes errors. Errors can be categorized in a number of different ways. They can be systematic or random. They can be intentional or unintentional. They can be serious or trivial. There are two fundamental rules regarding errors:

- Rule No. 1: All errors are made by humans.
- Rule No. 2: All errors are experienced by users.



[23]

Figure 2. [23] Discriminating the five verifications and validation.

There has never been an instance of an inanimate object making a mistake in medical device development, deployment, or maintenance. And users are not just end-users; they are also system testers, healthcare providers, hospital managers, maintenance personnel, and even patients. Figure 3 shows a different way of looking at errors that is more relevant to engineering systems.

Figure 3 separates errors into two categories (system use and individual user). It associates four types of human behavior (expected, unexpected, misguided, and malicious). The four different types of individual user errors are well-known: routine use, novel use, misuse, and abuse. Three of the four different types of system use errors (active, latent, and drift errors) are not so well-known outside of the HF community. The first two (active and latent errors) were defined by Reason and correspond loosely to "known bugs" and "unknown bugs", respectively.[7] The third (drift errors) was defined by Dekker and corresponds to an unintended transition of the system beyond its designed safety envelop (a drift towards failure).[3]

	ERROR CATEGORY	
ERROR TYPE	SYSTEM USE ERROR	INDIVIDUAL USER ERROR
Expected Behavior	Active (Known Bugs)	Routine Use
UnExpected Behavior	Latent (UnKnown Bugs)	Novel Use
Misguided Behavior	Drift (Beyond Design Envelope)	MisUse
Malicious Behavior	Sabotage	AbUse
	Locus of Control: Development, Deployment & Maintenance Organizations	Locus of Control: Individual Human(s)

[24]

Figure 4 shows how these three types of System Use errors might occur within medical device design control activities. Stakeholder dissonance (SD) is the condition where there is a conflict among the NWDs of different

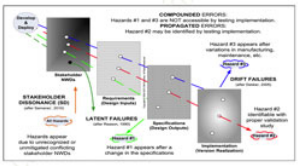
stakeholders; it is unrelated to cognitive dissonance [25].[12] SD may be thought of as a clash of values among stakeholders and is often characterized by errors, workarounds, threats to patient safety and organizational profitability, and even outright rejection of new technology. Managing SD is an essential element in the suppression of System Use errors. If a stakeholder, or one of a stakeholder's NWDs, is not recognized (or misinterpreted or ignored), the missing NWD will not appear in the design inputs, the engineers will not create corresponding design output(s), and, when the system is built, the propagated active error (Hazard #2 in Figure 4) may be detected as a system property during validation. So, while requirements verification may have failed (no requirement, no test ... seems like a pass, but error remains), in this case, validation may succeed, but only if it tests for that specific hazard! Unfortunately, this is not the case with compounded errors (Hazard #1 & #3 in Figure 4).

Figure 3. [24] **An error taxonomy. Adapted from [14].**

Figure 4 shows the critical limitations of system validation in the case of compounded errors.[9] If a hazard (a potential to cause harm) does not exist as a system property (is blocked and not expressed in the system version as built), when the validation is conducted, the hazard cannot possibly be detected.[6] Validation is based solely on testing the implemented design outputs (the current version of the system) against the documented, operationalized design inputs. If a design input is absent (either because the NWD was not recognized or it was recognized, but ignored), the system will incorrectly pass validation testing—because validation testing is only based on the design inputs. If specification verification fails, validation also fails and a subsequent change in specifications, or some manufacturing process, may then allow the blocked hazard (Hazard #1 in Figure 4) to appear in the final implementation. In the case of periodic revalidations after deployment, maintenance requirements that were not anticipated (another latent flaw) or that deviate from the original design (a drift towards failure, Hazard #3 in Figure 4) may also result in the system incorrectly passing validation. If implementation verification fails, validation also fails and a replacement part manufactured differently (e.g., from an alternate source) or a modified maintenance process (e.g., occurring at a different interval) may then allow the blocked hazard (Hazard #3 in Figure 4) to appear in the final implementation.

Opportunities for creating system use errors exist not only in the developing organization, but also in the deploying and maintenance organizations. This is an important distinction. The medical device manufacturer is not the only source of system use errors; the deploying and maintaining organizations may introduce new system use errors or may compound existing development-induced system use errors (see [14] for a simple health information technology example). Furthermore, while it is very simple to blame the end-user who, after all, was clearly the one that actually "did it" (remember Rule No. 2), this conclusion is logically flawed: end-user sanctions, counseling, retraining, or job redesign will not address system use errors [14].

So, how do we avoid exposing users to system use errors?



[26]

Figure 4. [26] **Error creation and propagation. Adapted from [9, 10].**

Mechatronics Maturity

Mechatronics maturity may be defined as: a measure of how good you are at avoiding or recovering from the creation or propagation of system use errors. Testing (verifications and validation) is an outcomes approach to quality; it is necessary, but not sufficient. Quality engineers understand this well. Inspection alone historically has proven inadequate and, in the 20th century, quality management moved successively through statistical quality control, quality assurance, and strategic quality management.[14] This is the process approach to quality practices; it is a management strategy that realizes that outputs are inextricably linked to inputs and transformations. The specific processes used by medical device development organizations can help us estimate their mechatronics maturity ... and that is a predictor of new product development (NPD) quality, which is one reason why FDA believes that quality system management audits have value.

As stated in the beginning, how an organization deals with the HW, SW, and HF issues may be viewed as a measure of mechatronics maturity. Table 1 shows one such means of assessing an organization's mechatronics maturity. As with other estimates of organizational maturity, it can be envisioned in five discrete stages.[15] From a goal-directed perspective, they range from uncertain (no clear goal) through tactical to strategic orientations. Organizational behavior ranges from ad hoc and chaotic all the way to a quantitatively-managed organization focused on continuous improvement. Consider the five stages of mechatronics maturity for a medical device development organization:

		MECHATRONICS MATURITY			
Maturity Stage	Organizational Behavior	Hardware (HW)	Software (SW)	Human Factors (HF)	Goal
5	Continuous Improvement	Human-Centered Systems Engineering			Strategic
4	Quantitative Management	Integrated HW-SW Systems Engineering		Stakeholder Dissonance Management	↑
3	Defined & Predictive	HW Systems Engineering	SW Systems Engineering	System Usability Engineering	Tactical
2	Qualitative Management	HW Engineering	Computer Science	Developer-Centered Usability	↑
1	Ad Hoc & Chaotic	Trial & Error	Computer Programming	Huh?	Uncertain

[27]

Table 1. [27] **Mechatronics maturity.**

Maturity Stage 1. Organizational behavior is ad hoc and chaotic. HW development is accomplished by trial & error. SW is the result of unstructured, free-flow, computer programming. There is no linkage between HW and SW development, resulting in frequent design conflicts and HW/SW incompatibilities. HF considerations are nonexistent or HF recognition is only very rudimentary. Product development errors are numerous; NPD is slow and expensive.

Maturity Stage 2. Organizational behavior is characterized by qualitative management of the engineering processes. HW development relies on the use of sound engineering principles and practices; testing is targeted and formalized. SW development focuses on language competencies and algorithm formulation. There is still no linkage between HW and SW development. HF is limited to usability and is primarily the domain of the individual HW or SW subsystem developer.

Maturity Stage 3. Organizational behavior is defined, documented, and proactive. Both HW and SW development follow the principles of classical systems engineering, but the managed processes are neither agile nor tightly linked. HF is still limited to usability, but now it is considered at the system level and is no longer determined by each HW and SW subsystem developer. Product development errors are well-controlled; the rate and cost of NPD is moderate.

Maturity Stage 4. Organizational behavior becomes characterized by quantitative management with non-intrusive, but detailed, performance data acquisition, analyses, and constructive feedback. HW and SW development are fully integrated and based upon systems engineering principles and practices. HF has gone beyond usability and encompasses the full spectrum of "cradle to grave" HF engineering with a focus on management of SD.

Maturity Stage 5. Organizational behavior continues to be characterized by quantitative management, but continuous improvement throughout the organization is emphasized in support of strategic goals. HW, SW, and HF development are formalized and fully integrated into a human-centered systems engineering process that continually considers all the stakeholders (and their organizations).[10, 13] Product development errors are infrequent; NPD is rapid and inexpensive relative to less mature competitors.

Conclusion

NPD presents a set of engineering management tradeoffs among the four basic NPD attributes: budget, schedule, scope, and quality. It is well-recognized that reductions in quality occur when organizations give greater priority to budget and/or schedule considerations.[1] But this is misguided. The correct objective should be to modify the NPD process to improve the stage of mechatronics maturity. A low maturity NPD process is more costly and less time efficient; trading off scope and quality reduce market share and increases potential liability. By contrast, a high maturity NPD process is quite cost and time efficient, permitting the developing organization to focus on maximizing scope and quality … and market share. So, even if your only motive is maximizing profit, not striving to maximize mechatronics maturity is misguided, counterproductive, and a failure of engineering management.

References

1. Dew J (2003). Root Cause Analysis: The Seven Deadly Sins of Quality Management [28]. Quality Progress.

2. Dew J (2011) Change Management: Tribal Quest [29]. Quality Progress.

3. Dekker SWA (2005). Ten Questions about Human Error: A New View of Human Factors and System Safety [30], New Jersey: Lawrence Erlbaum Associates.

4. FDA Report. Understanding Barriers to Medical Device Quality [31]. Oct. 31, 2011.

5. Grady JO (1997). System Validation and Verification [32]. CRC Press.

6. International Standards Organization (2007). ISO 14971§6.3: Application of risk management to medical devices: Implementation of risk control measures [33].

7. Reason J (1990). Human Error [34], Cambridge: Cambridge University Press.

8. Samaras GM & Horst RL (2005). A systems engineering perspective of the human centered design of health information systems [35], J. Biomedical Informatics, 38(1):61-74.

9. Samaras GM (2006). An Approach to Human Factors Validation [36], J. Validation Technology, 12(3):190-201.

10. Samaras GM (2010). Human-Centered Systems Engineering: Building Products, Processes, and Services [37]. Proc. 2010 SHS/ASQ Conference, February 25-27, 2010 on CD-ROM.

11. Samaras, GM (2010). The Use, Misuse, and Abuse of Design Controls [38]. IEEE Engineering in Medicine and Biology Magazine 29(3):12-18.

12. Samaras, EA & Samaras, GM (2010). Using Human-Centered Systems Engineering to Reduce Nurse Stakeholder Dissonance [39]. AAMI HF Horizons 2010; Biomedical Instrumentation & Technology 44(s1):25-32.

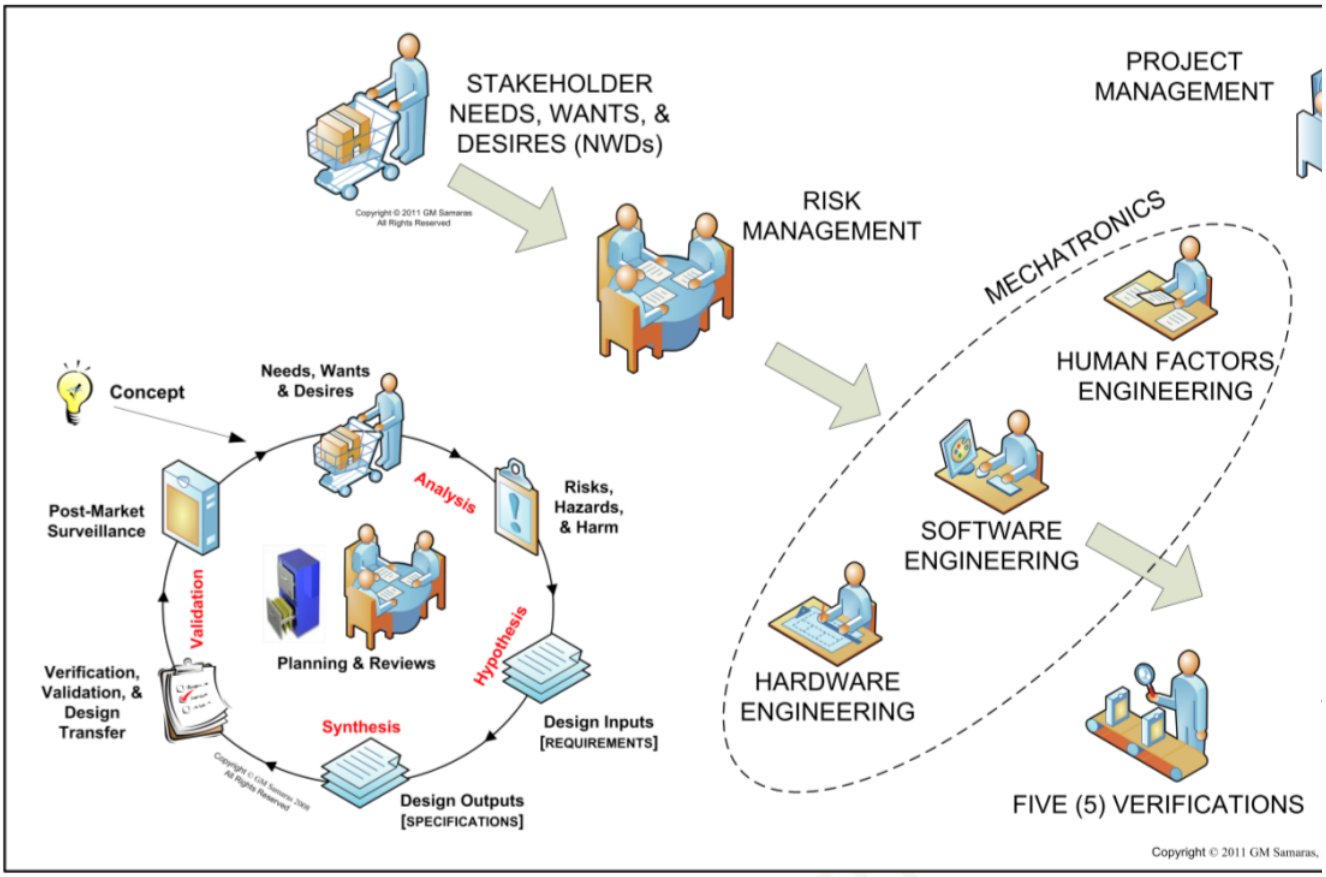
13. Samaras, G.M. Human-Centered Systems Engineering: Managing Dissonance in Healthcare Delivery, in Management Engineering for Effective Healthcare Delivery: Principles and Practices [40], Kolker, A. & Story, P. (Eds). Philadelphia:IGI Global pg. 148-171. 2011.

14. Samaras, GM. (2012). Reducing latent errors, drift errors, and stakeholder dissonance [41]. WORK: A Journal of Assessment, Prevention, and Rehabilitation, 41(s1):1948-1955.

15. Software Engineering Institute, Carnegie Mellon. Capability Model Maturity Integration. http://www.sei.cmu.edu/cmmi/ [42] Accessed 3/14/2012.

G.M. Samaras [43] is a biomedical scientist/engineer currently in private practice [44] (Pueblo, CO). Trained as an electrical engineer, he has doctorates in physiology and industrial engineering (engineering management); he is a licensed professional engineer (PE), board-certified human factors engineer (CPE), and an ASQ-certified quality engineer (CQE). He has a number of biomedical patents and publications in physiology and engineering (hardware, software, human factors, and quality). He has worked at the FDA/CDRH as a reviewer & manager, he was a medical school and engineering graduate school professor, and founded an engineering firm that he ran for a decade.

Author:
G.M. Samaras
Featured Image:



Design Feature Software Technology

Privacy Policy | Contact | Advertise | Subscribe | Sitemap
© 2013 UBM Canon

Related Sites from UBM Canon:

- Qmed - Qualified Medical Suppliers
- Medical Device + Diagnostic Industry
- European Medical Device Technology
- Medical Product Manufacturing News
- IVD Technology
- OrthoTec
- China Medical Device Manufacturer
- medtech*insider*
- medtech*insider* auf Deutsch
- Pharmaceutical & Medical Packaging News
- Pharmalive

Source URL: http://www.medicalelectronicsdesign.com/article/medical-device-mechatronics-maturity-0

Links:
[1] http://www.medicalelectronicsdesign.com/
[2] http://www.medicalelectronicsdesign.com/article/medical-device-mechatronics-maturity-0
[3] http://www.facebook.com/sharer.php
[4] http://www.medicalelectronicsdesign.com/categories/design
[5] http://www.medicalelectronicsdesign.com/department/feature
[6] http://www.medicalelectronicsdesign.com/categories/software
[7] http://www.medicalelectronicsdesign.com/categories/technology
[8] http://en.wikipedia.org/wiki/Innovation
[9] http://en.wikipedia.org/wiki/Mechatronics
[10] http://en.wikipedia.org/wiki/Hardware_abstraction_layer
[11] http://en.wikipedia.org/wiki/Stakeholder_theory
[12] http://en.wikipedia.org/wiki/Satisficing
[13] http://en.wikipedia.org/wiki/Engineering_management
[14] http://en.wikipedia.org/wiki/Systems_engineering
[15] http://en.wikipedia.org/wiki/Scrum_%28development%29
[16] http://en.wikipedia.org/wiki/Kanban
[17] http://en.wikipedia.org/wiki/Risk_management
[18] http://en.wikipedia.org/wiki/Operational_definition
[19] http://en.wikipedia.org/wiki/Requirements_engineering
[20] http://en.wikipedia.org/wiki/Concurrent_engineering
[21] http://www.medicalelectronicsdesign.com/sites/default/files/Samaras Figure 1.png
[22] http://en.wikipedia.org/wiki/Type_III_errors#Type_III_errors
[23] http://www.medicalelectronicsdesign.com/sites/default/files/Samaras Figure 2_0.png
[24] http://www.medicalelectronicsdesign.com/sites/default/files/Samaras Figure 3_0.png
[25] http://en.wikipedia.org/wiki/Cognitive_dissonance
[26] http://www.medicalelectronicsdesign.com/sites/default/files/Samaras Figure 4.png
[27] http://www.medicalelectronicsdesign.com/sites/default/files/Picture 6_0.png
[28] http://asq.org/pub/qualityprogress/past/0903/59theSeven0903.html
[29] http://asq.org/quality-progress/2011/12/basic-quality/tribal-quest.html
[30] http://www.leonardo.lth.se/sidney_dekker/books/ten_questions_about_human_error/
[31] http://www.fda.gov/downloads/AboutFDA/CentersOffices/CDRH/CDRHReports/UCM277323.pdf
[32] http://books.google.com/books/about/System_validation_and_verification.html?id=91Cwc-3YT-8C

[33] http://www.iso.org/iso/catalogue_detail.htm?csnumber=38193
[34] http://www.cambridge.org/gb/knowledge/isbn/item1132717/?site_locale=en_GB
[35] http://www.ncbi.nlm.nih.gov/pubmed/15694886
[36] http://www.gxpandjvt.com/ivtnews/templates/templateblank.aspx?articleid=1435&zoneid=18
[37] http://www.samaras-assoc.com/PublicDocs/PrePrint2010SHSASQ.pdf
[38] http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?reload=true&arnumber=5463000
[39] http://www.aami.org/publications/HFHorizons/pdfs/HF2010_Stakeholder_Dissonance.pdf
[40] http://www.igi-global.com/chapter/management-engineering-effective-healthcare-delivery/56252
[41] http://iospress.metapress.com/content/5278k3q3r1724220/?p=b40949bf480e447f8e4fa482c2f00fa7&pi=2
[42] http://www.sei.cmu.edu/cmmi/
[43] mailto:george@samaras.eng.pro
[44] http://www.samaras-assoc.com/