A Study of a Health Enterprise Information System,

Part 3 - Discussions with Software Performance Experts

Jon Patrick, PhD
Health Information Technology Research Laboratory
School of Information Technologies
The University of Sydney

Abstract

Evaluation of Clinical Information Systems (CIS) is usually performed by questionnaires administered to users to assess their level of satisfaction and usability of the system. In the modern enterprise system there is a layer of configuration of the system that requires a great deal of effort to engineer the customer's solution, which has not received scrutiny from researchers. Insight into the critical weaknesses and the consequences of the "configuration programming" of the Cerner Millenium software is provided through discussions with software quality and performance engineers whose job is to remedy the inefficiencies and concomitant insecurities created therein. The difficulties in maintaining Cerner sites for their customers are captured in the issues they have identified as weaknesses in the processing methods for creating a working CIS system.

1. Introduction

The debate on the effectiveness of EHR systems is contentious with few large scale studies able to identify the tangible economic or patient safety benefits. Yet there is a widespread perception that they should be effective in improving health care. This suggests that the measurements of effectiveness are not capturing the information that identifies the actual changes that are occurring on the ground. At the same time there is wide spread criticism of the performance of these systems for their user-unfriendliness and the unfit-for-purpose characteristics.

An aspect of the debate that has been rarely published is an appraisal of the technology from a technologists viewpoint. Obtaining this type of content is difficult because some vendors contractually require purchasers to agree to vendor policies that prohibit disclosure of HIT system design flaws and weaknesses, errors, bugs, and other hazards [1]. Only in 2010 have professional societies like the American Medical Informatics Association (AMIA) come out publicly against this practice which raises many ethical and business questions [2].

The perspective of software quality engineers has the possibility of shedding light on the technology design and implementation reasons that contribute to unsatisfactory performance, the limits of practicable performance, and the insights into the behaviour of users and software producers alike in a game of cat and mouse: giving the least for most price, and getting the most for the least price.

This essay is a record of discussions with a number of software quality experts who have performed assessments of working Cerner software in hospitals. They point to a range of technical issues that permeate through to the user interface and manifest deleteriously in ways that leave the user perplexed and frustrated. They also offer insights into potential ways to increase the system efficiency and usability.

2. The Comments of Software Validation Experts

The following 10 sections represent the commentary the software performance experts have made about their experiences of working with the Cerner Millenium suite over the last 10 years in the USA and Australia.

2.1. System Creation

The Cerner Suite of programs is highly complex with 3 integrated levels: storage consisting of 100s of tables, configurable parameters, and screen design. This complexity creates the following consequences:

1. It requires a large amount of time to learn it and understand the system's behaviour.
2. It is difficult to grasp the full consequences of any configuration decision due to interactions between the three levels. That is, it has now reached the state of a complex system.
3. The actual configuration design of the system comes from a Data Collection Worksheet (DCW). A DCW is Cerner's way of having the client fill in a spreadsheet which is then used to design the system.
4. Configuration updates are performed in one of three ways; through the Cerner build tools (such as DCLTOOLS), through a mass update Cerner build tool called BedRock, or through Cerner Control Language (CCL) scripts, which either take client-defined spreadsheets and import them or direct CCL (SQL-like) update actions. The same CCL script can contain references to different variations of its behaviour to be used for different hospitals. These scripts do not support any methods by which the behaviour can be validated as they are performed directly on the client system. A Cerner operation may work correctly for some time or for one orderable but not another. When a missing component is required for processing it can cause any of the following actions: total failure across one hospital, total failure across multiple hospitals if the code has a dependency (the one script can be written with built in differentiation for doing different things for different hospitals); no failure but loss of data, incorrect storage of data, and, incorrect data being stored or retrieved.
5. Removal/change of an order entry format (OEF). An OEF is part of the configuration setup and is a format designed to the customer's needs for creating some orders of materials or tests. Changing an extant OEF can remove functioning from some customer systems without warning and there is no method of getting that functioning back or fully accessing any orders that previously used that functioning part of the system. An example is when a designer creates an OEF to handle a specific set of medications then points those medications to that OEF. Later if that OEF is deleted from the system and there is no acknowledgement that it has been referenced, it leaves all of those medications pointing to something that no longer exists. This is regardless of whether a primary or secondary key is used to reference the order entry format.
6. Roll out consists of testing functions to perform in one scenario. All possible pathways e.g. the same function with different variations for different hospitals, are not tested, as there is an assumption that if it works for one pathway it will work for all.

2.2. Configurability

Configurability is both a need and potentially a curse in this case. It is clear that the massive level of configurability permits Cerner to unabashedly claim that they can deliver any configuration a customer requires. This is both true and false in that the potential is there but the reality is unattainable. The limitation is created by three things: the shortage of trained people skilled enough to construct the required design in the database system, the time span in which the deliverables are needed, and the money available to pay the people to create the required system. It is often found that the last of these proves to be up to three times the initial contract price, and organisations are unwilling to bail out of a project when they have expended a few hundred million dollars on it.

Fujitsu UK is an example of opting out when they believed the 900 Million dollars they had already expended would be less than what their future expense needed to be to reach any level of satisfactory performance.

The software experts reported that the Cerner database configuration maintenance requires ongoing and daily upkeep, but users are not advised of changes, nor it seems does anyone realize that a change in the database configuration tables can have wide sweeping effects on the whole system and are usually more severe than any code change. They report that in their experience that:

1. A database is normally thought of as having two dimensions; the schema and the data. The schema describes the tables and their structures. The data is information stored within those tables. Schema is static and data is dynamic. However, the Cerner system has a third dimension; configuration data within the tables. While the standard client data, such as orders on patients and the patient details, changes by the minute none of these changes affect the actual running of the system. The schema can change somewhat from site to site but is then generally static throughout that version of the software.

   More than half of the tables within the Cerner database are dedicated to storing 'build configuration'. The contents of these tables is what sets one hospital apart from another. It is how you can have the same version of the software running in a hospital in Sydney and in Seattle – both being radically different in terms of their usage. This is also the area of the system that can do the most damage and have the most effect on patient safety. A lot of products have areas which store configuration setup data. However, a Cerner Millennium system goes far, far beyond this. Minor updates to this configuration can have massive effects on the running of the system. For example, in Pharmacy just clearing one field in one medication can switch off drug interaction processing for that medication, or changing a single field elsewhere can now give an intern the ability to dispense cocaine. You would think that once defined this configuration data would be become static. However, in reality many people are changing many parts of this configuration daily. Most of the time Cerner personnel are making changes without the clients knowledge. So, it is not the schema that is changing but something far more dangerous.
2. The clients have little access to the appropriate Cerner documentation of the configurations for the purpose of learning or auditing, yet flaws in the design cause issues almost daily. However, due to missing investigations there is no way to point to one specific area other than to say it worked yesterday and it doesn't today, or this one works but that one does not and they were built the same. Such "issues" when reported are mysteriously corrected and no-one knows how. The response is "well, it works for us" and the client is left baffled and are told that they are at fault, leaving the customer mystified about failed tests on one day and passed tests on the next day. Once customers are educated in how to look for the changes to the system they can then take some control over this situation. With some queries written for the clients, we are able to arm them to help determine where this mysterious change was made and by whom. Even so, the hard evidence of change is often disputed by the Cerner person who made the change usually by saying "I just looked and did not make changes".
3. Some hospitals run daily logs over the system to identify what has been changed in the previous 24 hours and how it might effect them. However, these are limited and time consuming to verify.

The curse of configurability is multitude:

1. The user base asks for different variations of the system so that each new roll out has an increasingly more complex system to verify the impact of each new change;
2. The changes in the system become more and more dependent on the knowledge of the CCL programmers who have worked on the system, and their tasks increase in difficulty with added diversity and modifications;
3. The configuration process is designed so that initially the DCW spreadsheets are imported through a Cerner build tool which then populates the appropriate existing tables. However, after this import there are no more imports and the system is changed manually. At that point the tables can be adjusted if they wish. Not many installations do change tables but some do and they then adjust the pre-built SQL scripts to accommodate the change. The application servers then use these SQL scripts to collect and retrieve information and deliver it to and from the presentation layer. A system breaks after an upgrade when one of these modified SQL scripts are replaced during an update in the system, hence removing any client-specific setup within them; and,

4. The configurability functions do not have defined scopes that enable reliable prediction of how they will interact within other configurable functions, e.g. in one case the standard values of gender male/female/unknown were programmed into the data stores to have the values 1/2/3 respectively. A CCL

programmer changed the ordering sequence in the interface so that all males and females were switched in display order and therefore incorrectly reported in the presentation layer viewed by clinicians.

2.3. System Development Processes

Cerner's revenue from licensing has decreased and services increased over the last few years, which has shifted their development focus. The software experts surmise that the movement within Cerner's software engineering department has undergone a natural progression to more younger staff who are bringing experience with “sexier” interfaces, but without appropriate experience for the need to ensure veracity in data provided by external providers (e.g. medicines) and for patient episodes. This trend has expanded the "configurability", and hence complexity, at the expense of validation, and hence veracity.

There appears not to be any process that situates validation and design reliability ahead of "presentation inventiveness". For example:

1. The documentation is not of any great quality but at the same time the scale of the design is so great that consistency of meaning is extremely difficult to maintain, hence there is duplication and redundancy in the underlying storage structures that leads to inconsistent storage design decisions between one developer and the next. When these two designs come together then the data is not reliably stored and so not reliably retrieved.

2. The Entity-Relationship Diagrams (ERDs) are mainly used as a bargaining tool between Cerner and the client to document what was requested, what was suggested, and what was implemented. In cases when the client forces a feature against Cerner’s will then Cerner do have this to reference when that function breaks on the first update.

3. ERDs represent how the system was built and are classed as internal documents. Some clients get them and some do not. Therefore, this cannot be represented as “what was delivered”. They are informational reference guides. This second set of documents make references to the data model (the ERD’s in this case) and relate to the functionality requested by the client and Cerner’s response. Some are “edited” by clients who want changes and then there is a back and forth between Cerner and the client. The problem is that unreasonable requests by the client are eventually accepted which make the system a “one-off” and has a high probability of breaking when the next standard update to that area is released. The extent to which the ERDs form part of the contractual specifications between the vendor and purchaser are unclear.

4. There are no mechanisms in the Cerner suite to record the behaviours of the CCL programmers, hence it is not possible to identify the changes made by individual programmers, and so determine accountability.

2.4. Database Schema Management

The database schema are selected to suit a client’s specifications and then remain relatively stable, however maintenance activities on the tables can cause their own issues:

1. For Cerner facilities the client can opt to use RHO (Remote Hosting Option) or have a local install. In RHO environments Cerner hosts the back-end database, the servers for the processes and a Citrix farm to provide access. There is a WAN which connects this equipment with the hospital. These days about 60% of all Cerner installations are RHO. Local install is just that, where the client hosts all of the equipment. Secondly, regardless of which hosting option is chosen Cerner have complete control over the installation and database setup – even if the client eventually chooses to build their own configuration. Some access to the tables is through SQL stored procedures. When this is the case for certain tables the Cerner implementation team may change the schema slightly and then update the procedures for access. However, the problem then comes when those procedures are updated in some way through service packs. When this happens the system can break.

2. The schema documents are available to clients if they want to go out and dig for them. There is no "current" implementation since the Cerner associates installing each system have the ability to change the schema at will. Of course, this does not happen often but changes are made and some tables are used at one site and not another. They adjust built-in SQL scripts to accommodate the changes which means (hopefully) no changes to source code. The problem comes when the SQL scripts are updated and then replaced with the "master" copy and the system doesn't work after the upgrade.
3. Voluntary revelation of changes to the system would be tantamount to admitting that something was incorrect and if that change could be shown to have had an impact on some patient's health then the software manufacturer would be liable. Hence any system for bug revelation of software would destroy the current protection the software manufacturers have constructed through the positioning of the clinician as a 'learned intermediary' with all responsibility for the consequences of patient care. Hence, their opposition to publicly revealing errors and bugs.

2.5. The Handling of External Knowledge.

External knowledge is content that is collected elsewhere and embedded in the system by some form of loading process. Typical examples are drug tables, drug interactions, values and ranges associated with pathology tests, etc. The source of this knowledge can be from system internal repositories or external to Cerner. It is known that this knowledge has had errors in it from time to time either from the original source files or from bugs in the loading programs. For example some data is loaded from csv files which are generally known in the industry to carry faults that cannot be automatically filtered such as invalid values and missing values. In known cases values such as incorrect dosage have been identified, where the concentration rate, e.g. "Aspirin 325mg" has the dosage embedded in the name of the drug yet the field in the database where this value is stored is incorrect with a value of say, 1000mg.

2.6. Management of Orderables

Each orderable or procedure in the system is made up of multiple elements (such as the clinical category, order entry format, order alert information, etc.). A process for setting up the management of orderables has to be initiated for any system installation. In general there are between 20 and 27 elements that make up each orderable and require definition in the setup process. Most are required to successfully place an order, but not all. A situation can arise where the order by a clinician is accepted but due to missing elements in the set up, the order can disappear from the system (at a later date) without any trace or documentation. This can be dangerous as there is no evidence of the order so therefore no events or triggers on its completion are present, plus the clinical staff are left mystified or even in dispute as to whether the order was created and by whom in the first place.

Missing elements occur either when the orderable is being built originally (using the build tools or spreadsheet import), or the element is wiped out later by a maintenance revision using one of the same mechanisms. The clinician cannot cause the order to fail unless bad field values were entered originally in the setup process, when the definition of the order was loaded by the Cerner software. When bad field values are loaded and the clinician uses the orderable function then either, the application crashes, or shows an 'internal server error' message or allows the order that then fails later. However, most issues are caused when the orderable is incorrectly built.

Also, due to some of these potentially missing elements the order can be there and the clinician can complete it but then the completion will be reversed automatically. In one case several years ago we found a serious issue where completed orders were reverting to ordered but uncompleted. There was a situation where Insulin was ordered on a patient. The on-duty nurse gave the injection and then marked the order as completed. The next nursing shift came in later and logged in. The order showed pending again with an urgent flag (not completed) so the nurse gave the patient the dose of Insulin again. The patient suffered from an overdose of Insulin. This was an incorrect setup of the completion date and time format in the configuration.

Each of the processes for setting up the orderables systems effect the specific orders but not everything in the system, although one mistake in the order entry fields can affect every orderable that uses it.

2.7. Management of Medicines

The mechanism for identifying medicines uses an identifier, MULTUM. This acts as a pseudo primary key in the database tables. MULTUM is used to link the Cerner database of medications with an external MULTUM drug interaction database. There are NUMERIC and TEXT identifiers (IDs) across multiple tables that act as a link between the different tables to form the whole of the record about a medicine. As the IDs act in this way we would expect them to be declared as a Primary Key (PK) in the one primary table and then always referred to later in other tables as a Foreign Key (FK). This would ensure that the in-built functions of the Database Management System (DBMS) could be used to maintain the integrity of the drug identifiers. In particular that deletions to any element across the tables is checked for validity before it is executed. In the Cerner system we have found behaviours that suggest the key for linking is not set up as a PK-FK relationship so there is some doubt on the integrity of the full description of a MULTUM or the medication defined to be sustained by the DBMS. Also failure to maintain a correspondence between PK and FK values of IDs leads to broken links in the chain of description of any entry. Subsequently there could be incomplete information within the medicines and all orderable records when loaded into the customer application.

2.8. Maintenance Expenses

The process of testing revisions to a system is extremely expensive when one has to check the consequences of each given change to each hospital in a multi-hospital network. In the case of one hospital network they perform two roll outs per annum at a cost of $400,000 per roll out per hospital. This is done over 200 hospitals as each hospital's functionality has to be checked independently due to the single hospital specialisations that can exist within individual CCL scripts in response to configuration parameters. The net annual cost over 200 hospitals is $160 Million per annum.

2.9. Collaborations between Corporations

There was a situation in which iSoft had an established customer base for its iPharmacy system and Cerner had to introduce a collaborating system. Cerner insisted that iSoft conform to a number of requirements, including that iSoft:

- convert its Australian Medicines Terminology coded pharmacy items to Cerner's proprietary MULTUM code numbers. This meant that iSoft had to buy access from Cerner to the MULTUM codes and build and maintain the mapping tables despite operating with the Australian standard.
- messages would be sent by iSoft as HL7 messages V2.4.1 which Cerner would convert to V2.1 formats for processing, and likewise use V2.1 for sending messages to the iSoft system. Cerner also would not acknowledge the receipt of messages.
- Cerner only update their medicines every 6 months while iSoft update theirs every three months

iSoft determined Cerner were using older technologies and lesser standards to their own operations and that this would have been too much risk clinically and declined the proposals, subsequently iSoft was positioned as the obstructor in the process (personal communication, ex-iSoft employee).

2.10. Site Specific Case Studies of Interest

2.10.1. Potential Failure of HealthSmart, Victoria

During the HealthSmart project in Victoria, Australia, PowerChart was to be installed and the customer sought to use ill-advised combinations of many different products. This was brought about by the lead team being inexperienced in Health Information Technology tasks and equally of limited clinical experience and without effective clinical leadership or champions. Cerner advised against many of the proposed adaptations but agreed to do them at the "customer's insistence". It is predicted that this system is now so convoluted and poorly designed that it will never be rolled out or will fail upon the first bug fix release of the

standard product - current investment ~ $300million and there is no practical implementation after 5 years of effort.

2.10.2. Reporting in Cerner.

The Cerner reporting module is a menu list of composed CCL scripts. New reports are created by writing new CCL scripts. The CCL is extremely powerful and can cause significant harm if used inappropriately, for example it allows for the removal of any data. The CCL has update capabilities along with report/query capabilities in a single script.

NSWHealth refuses to produce reports from the Cerner EMR as needed by ED Directors. Refusal to create reports is an unwillingness to write the scripts and/or pay the money to Cerner to have them written. It can also indicate that there are not enough staff available for the CCL programming process. Most clients in the USA go to 3rd party consulting companies with CCL experts to produce the reports required. In general, this is cheaper and faster than using Cerner.

3. Analysis

The perspectives of the software performance experts produce a different explanation to what we have seen in past studies of HIT systems. Instead of painting a picture of unhappy users plying away at a user interface that makes their task twice as difficult they understand the difficulties to originate more deeply in the process of software production. The problems arise in weaknesses that are introduced due to software engineers with limited understanding of the nature of the task to be completed and the skills required to address them.

System complexity is not treated as a factor in clinical systems assessments. These experts believe that the intrinsic complexity of the system emanating from its pure scale and its diversity arising from its configurability makes the system non-deterministic in its own behaviour. It also makes the system incomprehensible to the clinicians who have to use it as a tool to configure systems for others in a domain they have little expertise in and with mostly only youthful endeavour to push them along. Hence the requirement to have a reliable (read deterministic) technology is intrinsically destined to fail. The configurability of the system is shown here to be so diverse that mistaken processing is bound to be created in the configuration stage. A formally defined validation process is required to ensure a process operates correctly and does not lose data, and properly record and present to the user the entry of an order and the completion of its execution. Added to this observation is the further complication of the move to Internet technologies which by their nature are designed to be non-deterministic, that is, there is no guarantee that all packets sent from a server will reach their destination, hence that a system will have all the information it needs to have to deliver any complete service to its users.

Further to the complexity problem is the belief by the experts that the vendor has less than optimum strategies for supporting users when bugs are detected in the system, that is, dealing with bugs by fixing them then discounting their previous presence. We would have been unable to see this strategy through an information collection process solely relying on user interviews. It requires technical experts to understand the processes of building and repairing code to detect these practices.

The experts also provide an insight into a process that is very much taken for granted, that of loading external knowledge into the operational system. In this case important aspects of weaknesses in this process are presented by the experts in that incoming data is not validated for its own intrinsic correctness and consistency. They also have hints that the use of the intrinsic checking mechanisms available in the database management software, that is primary and foreign key checking, is over-looked allowing the possibility of further mistakes in data to creep into the operational system.

Validation of data in external knowledge sources needs to be the responsibility of any organisation that installs such data into its systems. This could take a number of methods: initial capture of the data needs to be done by double entry keying with the computation of key stroking similarity scores and thereby

concomitant computation of the error estimates of the content. Storage of the information should be in database management systems that provides for automatic checking of domain ranges and primary key values and foreign key references. Released clinical information systems should maintain records on when they have been loaded from the certified versions of a knowledge database.

More details of the difficulties of the configuration system are highlighted with the provision of the descriptions of how orderables are specified. The complexity of creating an orderable is provided giving added insight into how readily a configuration programmer could make a mistake in defining one. However, not only are the ramifications of making a mistake serious, but also they point to the difficulty clinical staff would have in understanding the behaviour of their system as just entering an order could cause the system to crash. This of course would be a more fortunate scenario for the patient as it prevents the system triggering actions that would be deleterious as demonstrated in the case where it directly caused an overdose of insulin for a patient.

The experts describe the process of validating revisions to any system and its ultimate cost to an organisation. The cost is seemingly excessive when viewed on the scale of a statewide hospital system of over 200 hospitals. The emergence of large enterprise systems that are of such a complex scale indicates a need for processes that audit the actions of the designers in configuring a system just as the IT industry has formulated for the processes of data entry, retrieval, and code reviews.

The disadvantages of using non-standard system descriptions is illustrated in the pharmacy system when a foreign system was introduced amongst an Australian standardised system. This however has a deeper interpretation. Cerner positioned themselves as the controlling authority in the Pharmacy project and required other participants to conform to their systems behaviour. This gives them two advantages: others have to carry the cost of conforming to their standards, and without external standards their own systems do not have to conform to any particular standards making it more difficult for other vendor systems to collaborate with them effectively. This enables Cerner to characterise the other vendors as the obstructionists in the process. It is interesting that the basis of the differences is that Cerner was using an older standard, unconventional processing, and less appropriate classification than iSoft, thus positioning itself as technologically inferior, yet at the same they were prepared to assert that other parties should have conformed to their technology.

In the case of Victoria's HealthSmart project we learned that projects can also be derailed by the behaviour of a client who doesn't use the appropriately skilled people to drive the technology configuration. This has come with great cost incurred without having produced effective deliverables over a 5 year period.

The experts give an explanation of the report writing technology. The Cerner Command Language is revealed where report writing is a straightforward process and so that the failure to supply reports might be driven by a lack of skilled staff to create the wanted scripts.

4. Conclusions

The experience of software performance experts working with Cerner systems having the ability to recognise weaknesses in its architecture and limitations of its usability offer a new founded insight not reported in the HIT literature. They are able to illuminate inner machinations of the technology that are not thought of as pertinent to technology appraisal. They are clearly able to offer richer and more explanatory causes for weaknesses and limitations in the usability of HIT systems than any user focused usability survey.

References

1. Koppel R, Kreda D. Health care information technology vendors' "hold harmless" clause: implications for patients and clinicians. JAMA 2009; 301:1276e8.

2. Kenneth W Goodman,1 Eta S Berner,2 Mark A Dente,3 Bonnie Kaplan,4 Ross Koppel,5 Donald Rucker,6 Daniel Z Sands,7,8 Peter Winkelstein,9 for the AMIA Board of Directors. Challenges in ethics, safety, best practices, and oversight regarding HIT vendors, their customers, and patients: a report of an AMIA special task force. J Am Med Inform Assoc (2010). doi:10.1136/jamia.2010.008946