

# Data Quality and Systems

ONE CERTAIN WAY  
TO IMPROVE THE  
QUALITY OF DATA:  
IMPROVE ITS USE!

In the movie *War Games*, set in the U.S. Strategic Air Command Center under a mountain near Colorado Springs, there is a critical scene: on a

huge map of the globe are arrows indicating large numbers of nuclear missiles approaching the U.S. from the Soviet Union. The military official in charge is trying to decide if he should ask the President for permission to launch a retaliatory attack because the technicians are telling him the threat is real. At this point, the scientist who originally designed the system (and who knows that something is wrong with the system itself) says, “General, what you see on that board is not reality, it is a computer-generated hallucination!”

**R**ecently, a former Director of the CIA revealed that a real-life version of this fictional scenario was actually played out when a test tape was inadvertently installed and the screen at a similar command center warned of a similar nuclear attack. As computers play increasingly important roles in the real world—a world in which computer-generated outputs often present a picture of the real world for critical

activities—it is increasingly vital that the pictures being displayed are correct!

In the mid-1970s, a number of my colleagues and I developed a model for information systems that predicted: (1) major computer systems problems involved with making the transition to the year 2000; (2) data quality difficulties in many operational systems being developed at the time; and (3) fundamental issues involved in the accuracy of

# Theory

confidential/secret data [1].

The theory that allowed us to formulate our predictions involved viewing information systems as subsystems embedded in a larger framework of a real-world feedback-control system (FCS) (Figure 1). Two observations from our work caused us to look at information systems this way: (1) all of the information systems we developed operated in a larger, goal-seeking, organizational environment, and (2) those systems that failed to take into account a larger FCS context were difficult to operate and their outputs were difficult to reconcile with the real world. We began to see that the data and data quality in our information systems did not exist in a vacuum. As a result, we began to explore the implications of a true systems (cybernetic) model for information systems.

The principal role of an information system is to present views of the real world so that the people in an organization can create products or make decisions. If those views do not substantially agree with the real world for any extended period of time, then the system is a poor one, and, ultimately, like a delusional psychotic, the organization will begin to act irrationally.

From the FCS standpoint, data quality is actually quite easy to define. Data quality is the measure of the agreement between the data views presented by an information system and that same data in the real world. A system's data quality of 100% would indicate, for example, that our data views are in perfect

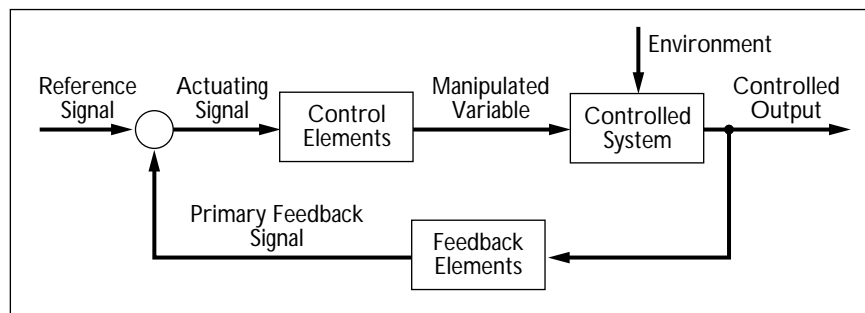


Figure 1.

Feedback-control systems model

agreement with the real world, whereas a data quality rating of 0% would indicate no agreement at all.

Now, no serious information system has data quality of 100%. The real concern with data quality is to ensure not that the data quality is perfect, but that the quality of the data in our information systems is accurate enough, timely enough, and consistent enough for the organization to survive and make reasonable decisions.

Ultimately, the real difficulty with data quality is change. Data in our databases is static, but the real world keeps changing. Even if our system has a database that is 100% in agreement with the real world at time  $t_0$ , at time  $t_1$  it will be slightly off, and at time  $t_2$  it will be even further off. FCS theory states that if a system is intended to track the real world, there must be a mechanism to synchronize the data in the system with changes in the real world—feedback is necessary!

But where does this feedback come from? The classic answer from information systems developers is that feedback is solely the responsibility of the users of the

system. “Our job is not to understand what our systems are being used for or even their context!” the systems developers maintain. “We simply build systems that meet the requirements of our users—it is the job of the users to ensure that the data in our databases is maintained in an accurate and timely manner. The best we can do is to ensure that the database is internally consistent and that the users’ business rules are enforced.”

Users, on the other hand, historically have felt that they were held responsible for data quality in information systems that they often did not understand, systems in which it was often difficult to make appropriate corrections, and systems in which the results of certain kinds of changes were unpredictable.

Unfortunately, the problem of data quality is fundamentally intertwined in how our system fits into the real world; in other words, with how users actually use the data in the system. Two things have to happen for data in any database to track the real world: (1) someone or something (a person or an automatic sensor) has to compare the data views from the system with data from the real world, and (2) any deviations from the real world have to be corrected and reentered.

Too often, systems developers have an overly sim-

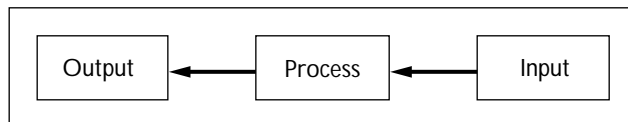


Figure 2.  
Input-Process-Output model

plistic view of how systems are organized; they think of systems in a simplistic Input-Process-Output (IPO) “transform” model (Figure 2). But, the IPO model fails to account for the role that the database plays in a broader context, as depicted in Figure 3.

In real information systems, the database acts to mediate<sup>1</sup> between the input and the output, where the input and output: (1) occur at different times, and/or (2) represent different views of the real world. This broader view of a system then makes it possible to understand fully the FCS model, in which the information system fits within actions taken in the real world (see Figure 4).

In the FCS model, data is entered in the system based on external inputs. It then undergoes processing and is stored in a database, which in turn is processed to produce outputs that are used in (compared with) the real world. Finally, new inputs are produced (and fed

back) so that the database remains accurate. Without this final loop, the system will fail to maintain its database and its outputs correctly. This final FCS model (Figure 4) allows us to understand more fully the true problem of data quality—the better our information system fits into the real world, the better the quality of our data will be, the worse the fit, the worse the data.<sup>2</sup>

## Data Quality Rules

There are a number of general data quality rules one can deduce from a FCS view of information systems:

- DQ1. Unused data cannot remain correct for very long;
- DQ2. Data quality in an information system is a function of its use, not its collection;
- DQ3. Data quality will, ultimately, be no better than its most stringent use;
- DQ4. Data quality problems tend to become worse as the system ages;
- DQ5. The less likely some data attribute (element) is to change, the more traumatic it will be when it finally does change;
- DQ6. Laws of data quality apply equally to data and

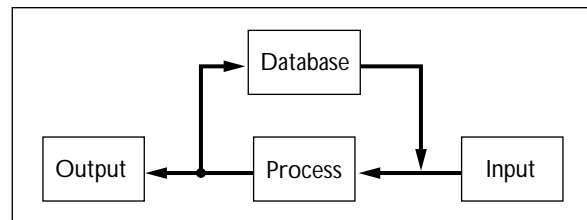


Figure 3.  
Input-Process-Database-Output model

metadata (the data about the data).

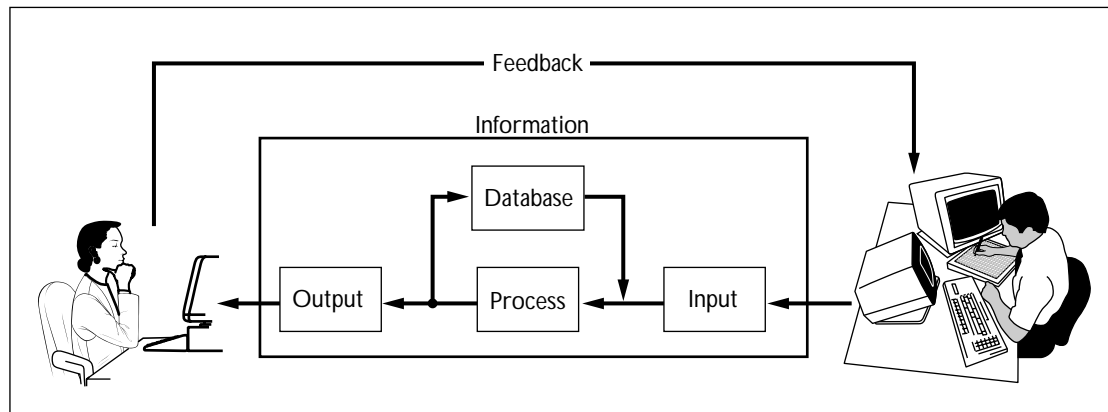
**Data Quality: Use It or Lose It.** Unfortunately, many, if not most, of these data quality rules fly in the face of traditional systems practice. It is common practice to collect large numbers of unused data elements on the premise that someday someone might want to use them, and that it is cheaper to put them in the system now than to do so when the need actually arises. However, applying the FCS model, it is clear that if an organization is not using data, then, over time, real-world changes will be ignored and the quality of the data in the system will decline.

In biological systems, scientists refer to this phe-

<sup>1</sup>If the inputs and the outputs of a system were in the same sequence and there was no time differential, there would be no need for a database at all. Like the human memory, the database mediates between information needs that occur at different times or in different sequences.

<sup>2</sup>It is possible to formulate other ways in which data in a database can deviate from the real world: poor data definitions, failure to correctly enter data, rounding and/or compounding errors in calculations and faulty calculations. However, while these other errors can create data quality problems, those created by lack of consistent user feedback dwarf all other kinds of errors.

**Figure 4.**  
Information systems in a real-world context



nomenon as atrophy—if a part of the body is not used, it atrophies. In a practical sense, something similar to atrophy happens with unused data—if no one uses the data, the system becomes insensitive to that data. As with an individual who is blind or deaf, certain changes in the real world are not perceived and registered.

In most large systems, however, it is difficult to tell if particular data elements are actually being used. For example, some poor quality data elements appear on reports or screens but no one actually uses those reports or screens. In other cases, the data elements are used, but not very seriously. Here DQ3 comes into play, namely, that the quality of a specific piece of data will be no better than its most stringent use. In general, the quality of data that is not stringently used will be better than data that isn't used at all, but not much better. For example, names and addresses that are used only for mailing lists and are not corrected based on returned mail tend not to be very accurate.

The nature of data quality hinges upon the connections of that system to the outside world. The stronger those connections, the better the system and the better the data quality.

**Data Warehousing and Data Quality.** Recently, as large organizations have begun to create integrated data warehouses for decision support, the resulting data quality problems have become painfully clear [1]. These organizations have discovered that the quality of the data in their legacy databases is their single biggest problem. One data manager for a large company reported that fully 60% of the data transferred to their data warehouse failed to pass the business rules that the system's operators had said were in force, something that could have been predicted based on poor data usage.

On the plus side, the mere development of data warehouses represents a quantum leap forward in terms of end-user usage of data. As more advanced data warehouses and data marts are created, more people will be using data in more stringent ways. The need for quality data has already begun to focus more management attention on the poor state of most data quality.

In the 1970s, when my colleagues and I first began to understand the implications of the FCS model, it became clear why so many of the systems we had worked on failed to meet their data quality objectives. We had developed systems that created data that no one used. We recognized that such systems were difficult to define, difficult to program, and difficult to operate—but we didn't understand why.

Using the FCS model, we created a new development approach that helped ensure that all data collected and stored would actually be used. That approach involved designing systems by working backward from uses to outputs to database to inputs in a controlled fashion [2]. In one case, we found the legacy system being replaced had three times more data elements than it actually needed. Imagine the problems with data quality. Attempting to build quality systems without understanding FCS theory is much like attempting to build an airplane without understanding aerodynamics.

**Data Quality and the Year 2000.** We knew that the coming of the year 2000 would create a serious problem because there would be very little use of the millennium and century fields until the year 2000 actually arrived.<sup>3</sup> Unfortunately, we failed to see just how massive the problem would actually be. The real problem involved with finding, fixing, and testing the changes for the year 2000 dilemma is not its difficulty but its ubiquity—a simple problem repeated a billion times.

Could the year 2000 problem have been avoided? Possibly, but only if some universal form of "use-based" data quality had been put in place. Millennium and century fields have not been tested on a large scale because of the "time horizons" of these systems do not yet use those fields in ways sensitive to the turn of the century. As the year 2000 approaches, more and more systems will fail because their systems practice will be

<sup>3</sup>Some organizations have had to deal with this for decades. For example, savings and loan companies financing 30-year mortgages have had to deal with the year 2000 since the late 1960s. For the most part, organizations are only now reaching their "century time horizons" where they need accurate dates that reach into the 21st century.

forced to use dates that occur in the 21<sup>st</sup> century.

**Data Quality, System Age, and Metadata.** As systems get older their data quality problems tend to worsen. In the 1960s and 1970s, it was widely thought that the lifespan of the average information system would only be a few years. In this context, it didn't make sense to try to install costly data quality programs, since any problems or shortcomings in the current system would be corrected in subsequent versions. In fact, major information systems have turned out to be much longer lived than anyone would have anticipated. There are large numbers of legacy systems in operation today that date back 20 or 25 years. Consequently, it is necessary to assess the impact of time on data quality.

Not only does data quality suffer as a system ages, so does the quality of its metadata. What happens is that people who are responsible for entering the data discover which data fields are not used; either they then make little effort to enter the correct data, or they begin to use the data for other purposes. The consequence is that both the data and the definitions of the data (the metadata) no longer agree with the real world.

Another predictable problem occurs when the data model used in systems differs significantly from the real world. In such cases, the structure of the data in the system no longer agrees with the current structure of the business. Typically, systems designers do not actually look at the structure of data that occurs in various fields, but rather arbitrarily assign data to fixed fields based on technology limits or constraints. Because the developers are not noticing changes in the data structure, the structure is not changed, and as a result, round data is forced into square fields.

**Data Quality and Secrecy.** One of the most troubling implications of the FCS model to data quality has to do with confidentiality and secrecy. If the quality of data is truly wrapped up in its use, then there seem to be serious limitations to the quality of confidential/secret data. One implication seems to be that confidential/secret data will always have limited quality. This may account for the fact that while dictatorships seem to be an efficient way to run a society, democracies, for all their inherent inefficiencies, work better. A free press and an open political process, though sometimes bothersome, provide feedback and therefore keep data quality high.

**Data Quality and Information Overload.** It is currently not clear what impact information overload will have on data quality. In our modern technologically-based society, if we are to preserve quality in our key data, there may be such a thing as too much data. Clearly, one of the consequences of such enormous amounts of data being available is the difficulty in find-

ing important data and being able to compare similar data from different sources.

## Use-based Data Quality Programs

If data quality is a function of its use, there is only one sure way to improve data quality—improve its use! We call this *use-based data quality*. Use-based data quality programs are built around finding innovative, systematic ways to ensure that critical data is used.

**Use-based Data Quality Audits.** To improve our data quality, it is necessary to determine how good the data in our databases is today. Use-based data quality audits involve answering a number of key questions:

*What data are we interested in?*

*What is the data design?*

*What is the data model?*

*What is the metadata?*

*How is the data used today?*

*Who uses it?*

*For what purposes is the data used?*

*How often is the data used?*

*What is the data quality?*

*What is in the database?*

*How does it compare with the current data in the real world?*

*How current is the data?*

For the most part, data quality audits are best done using statistical sampling. It is rarely a good idea to try to verify all of the data in a real database; it is necessary to create a sufficient sample that will enable us to draw meaningful conclusions.

**Use-based Data Quality Redesign.** To improve data quality, it is mandatory to improve the linkage among the various uses of data throughout the system. One of the problems is deciding where to begin. While most legacy environments contain hundreds of records (tables) and thousands of data elements, all the data is not equal. In most systems, there are a few critical sets of data that make all the difference. Often, the “customer,” “product,” “order,” and/or “organizational structure” categories of data are most important. The first step in a serious data quality redesign program is to identify the critical data areas. This involves a careful reexamination of how critical pieces of data are used. Normally this is manifest in two areas—the basic business processes (order entry through fulfillment, for example), and decision support. Use-based design means focusing on exactly how the data will be used, and in trying to identify inventive ways to ensure that the data is used more strenuously. In many cases, this means creatively persuading the people most knowledgeable about the data to take responsibility for it.

A good example is the frequent flyer programs offered by the airlines. In addition to creating customer loyalty, such programs also go a long way to improving the quality of the airlines' data. In case the same flyer may have more than one assigned frequent flyer number, it is in the best interest of the customer to make sure that the records are consolidated and vital information such as name, address, family relationships, and preferences remain current and accurate. The best kind of data quality program is one in which the data subject has a vested interest in keeping the data correct!

Developing a use-based data quality program requires much more effort in the actual process of completing the feedback use of data. In general, that means reducing the number of data elements collected. If data cannot be maintained correctly, then one should ask whether that data provides any value to the enterprise.

Another major component of use-based design is to understand the content of critical existing databases. A number of tools have emerged in recent years that attempt to analyze and combine data from multiple databases to create a common view of "customers," "products," and "vendors," for example. The normal result of these programs is to dramatically reduce (consolidate) the size of major databases. Consolidations of 5:1 or even 10:1 are not uncommon. A second byproduct derived from this process is the development of a much more sophisticated set of metadata based on data content.

Another technique for improving data quality is to promote (demand) "sharing" data through the use of "common databases." With the advent of the Internet, more and more people are able to access data easily. Providing easy data access to a broader audience has the long-term effect of dramatically improving data quality.

**Use-based Data Quality Training.** One of the major problems with data quality is getting both users and managers to understand the fundamentals of data quality. In order for any data quality program to work over the long term, a significant amount of time must be devoted to education and training. It is hard to convince users and managers who have been used to requiring data without thinking about its use that much of what they are collecting will never be correct. Fortunately, people become converts in a relatively short time when they begin to see the effects of a conscientious data usage program.

**Use-based Data Quality Continuous Measurement.** Data quality requires constant measurement to ensure that use-based practices are followed throughout various processes. As Deming noted, most quality problems are systems problems, not worker problems.<sup>4</sup>

<sup>4</sup>Mary Walton quotes Deming as saying "Workers work within a system that—try as they might—is beyond their control. It is the system, not their individual skills, that determines how they perform [3]."

However, individual errors also contribute to poor quality data. Measurement and quality programs must go hand-in-hand. All the questions that were raised in the data quality audit need to be regularly repeated for the redesigned system as well.

A final note on measurement: Do not be persuaded that internal measures without external verification are adequate. All that internal measurement can do is ensure that data is internally consistent. No large organization can rely on its inventory records without periodic "physical inventories." Having records that show that there should be 23 computers in Warehouse X does not mean that there are actually 23 computers on the shelves. If we want the data residing in our databases to agree with the real world, we must periodically verify that those computers actually exist where our system says they exist, and we must take actions to reconcile any differences. Data that is truly vital must be physically audited.

## Conclusion

Too often, the primary focus of data quality projects is to increase the internal controls involved in entering and editing data. As laudable as these efforts are, they are ultimately doomed to failure, as are one-shot attempts to clean up data. The only way to truly improve data quality is to increase the use of that data. If an organization wants to improve data quality, it needs to ensure that there is stringent use of each data element.

Because of the potential for year 2000 problems, every organization in the world that uses computers will have to confront the problems of data. This, coupled with the increased need for quality data for decision making, will make data quality a high priority item in every enterprise. Use-based data quality provides a theoretically sound and practically achievable means to improve data quality. ■

## References

1. Orr, K. *Structured Requirements Definition*. Ken Orr and Associates, Topeka, KS, 1981.
2. Redman, T.C. *Data Quality for the Information Age*. Artech House, Boston, MA, 1996.
3. Walton, M. *The Deming Management Method*. Perigee Books, NY, 1986.

---

**Ken Orr** (kenorr@kenorrinst.com) is a principal researcher at the The Ken Orr Institute, a business technology research organization in Topeka, KS.

---

Permission to make digital/hard copy of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.