CHAPTER 5

Data Quality Issues Management

Data quality investigations are all designed to surface problems with the data. This is true whether the problems come from stand-alone assessments or through data profiling services to projects. It also does not matter whether assessments reveal problems from an inside-out or an outside-in method. The output of all these efforts is a collection of facts that get consolidated into issues. An issue is a problem with the database that calls for action. In the context of data quality assurance, it is derived from a collection of information that defines a problem that has a single root cause or can be grouped to describe a single course of action.

That is clearly not the end of the data quality effort. Just identifying issues does nothing to improve things. The issues need to drive changes that will improve the quality of the data for the eventual users.

It is important to have a formal process for moving issues from information to action. It is also important to track the progress of issues as they go through this process. The disposition of issues and the results obtained from implementing changes as a result of those issues are the true documentation of the work done and value of the data quality assurance department.

Figure 5.1 shows the phases for managing issues after they are created. It does not matter who performs these phases. The data quality assurance department may own the entire process. However, much of the work lies outside this department. It may be a good idea to form a committee to meet regularly and discuss progress of issue activity. The leader of the committee should probably be from the data quality assurance department. At any rate, the department has a vested interest in getting issues turned into actions and in results being measured. They should not be passive in pursuing issue resolution. This is the fruit of their work.

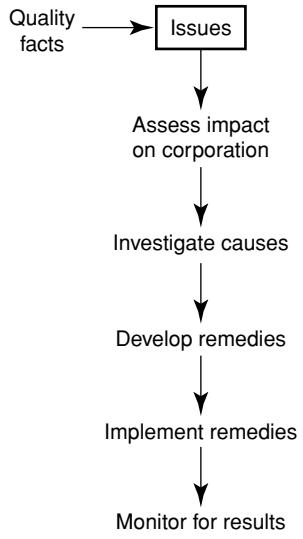

FIGURE 5.1 Issue management phases.

An issue management system should be used to formally document and track issue activity. There are a number of good project management systems available for tracking problems through a work flow process.

The collection of issues and the management process can differ if the issues surface from a "services to project" activity. The project may have an issues management system in place to handle all issues related to the project. They certainly should. In this case, the data quality issues may be mixed with other issues, such as extraction, transformation, target database design, and packaged application modification issues. It is helpful if data quality issues are kept in a separate tracking database or are separately identified within a central project management system, so that they can be tracked as such. If "project services" data profiling surfaces the need to upgrade the source applications to generate less bad data, this should be broken out into a separate project or subproject and managed independently.

5.1 Turning Facts into Issues

Data quality investigations turn up facts. The primary job of the investigations is to identify inaccurate data. The data profiling process will produce inaccuracy facts that in some cases identify specific instances of wrong values. Other cases identify where wrong values exist but identification of which value is wrong is not known, and in yet other cases identify facts that raise suspicions about the presence of wrong values.

Facts are individually granular. This means that each rule has a list of violations. You can build a report that lists rules, the number of violations, and the percentage of tests performed (rows, objects, groups tested) that violated the rule. The violations can be itemized and aggregated.

Metrics

There is a strong temptation for quality groups to generate metrics about the facts and to "grade" a data source accordingly. Sometimes this is useful; sometimes not. Examples of metrics that can be gathered are

- number of rows containing at least one wrong value
- graph of errors found by data element
- number of key violations (nonredundant primary keys, primary/foreign key orphans)
- graph of data rules executed and number of violations returned
- breakdown of errors based on data entry locations
- breakdown of errors based on data creation date

The data profiling process can yield an interesting database of errors derived from a large variety of rules. A creative analyst can turn this into volumes of graphs and reports. You can invent an aggregation value that grades the entire data source. This can be a computed value that weights each rule based on its importance and the number of violations. You could say, for example, that this database has a quality rating of 7 on a scale of 10.

THE GOOD

Metrics can be useful. One use is to demonstrate to management that the process is finding facts. The facts have little to no significance by themselves but can be circumstantial evidence that something is wrong with the data. When a data quality assurance department is trying to gain traction in a corporation, metrics can be a useful way to show progress.

Metrics can also be useful to show improvements. If data is profiled before and after corrective actions, the metrics can show whether the quality has improved or not.

Another use of metrics is to qualify data. Data purchased from outside the corporation, such as demographic data, can be subjected to a quick data profiling process when received. Metrics can then be applied to generate a qualifying grade for the data source. It can help determine if you want to use

the data at all. This can be used to negotiate with the vendor providing the data. It can be the basis for penalties or rewards.

Qualification can also be done for internal data sources. For example, a data warehousing group can qualify data extracts from operational groups before they are applied to the central data warehouse.

THE BAD

The downside of metrics is that they are not exact and they do not solve problems. In fact, they do not identify what the problems are; they only provide an indicator that problems exist.

Earlier chapters demonstrated that it is not possible to identify all inaccurate data even if you are armed with every possible rule the data should conform to. Consequently you cannot accurately estimate the percentage of inaccuracies that exist. The only thing you know for sure is that you found a specific number of inaccuracies. The bad news is that there are probably more; the good news is that you found these. If the number you find is significant, you know you have a problem.

Corrective actions have these potential consequences: they can prevent recurrence of some errors that you can detect, they can prevent recurrence of errors you cannot detect, and they can continue to pass errors through. It is also theoretically possible that you would introduce new errors that may or may not be detectable.

The conclusion is that data profiling techniques can show the presence of errors but cannot show the absence of errors nor the number of errors. Therefore, any metrics derived from the output of profiling are inexact. This does not make them useless. On the contrary, the errors found are true errors, and if there are enough of them you have uncovered true problems.

You might conclude from the previous discussion that the number of errors reported is understated. This would be great if it were true. However, poorly defined metrics can actually overstate the error condition. This occurs when a single inaccurate value triggers multiple rule violations. This is difficult to detect and impossible to quantify. When you consider that the majority of rules will find the presence of inaccurate data but will not pinpoint the offending values, you can see why it is difficult, if not impossible, to find the true number of inaccurate values.

Comparing metrics can also be misleading if the yardstick changes between profiling exercises. As analysts gain more knowledge about a data source, they will add to the rule set used to dig out inaccuracies. Comparing two result sets that are derived from different rule sets results in an apples-to-oranges comparison. All presentations of quality metrics need to provide disclaimers so that the readers can understand these dynamics.

THE *following is an example of preventing recurrence of errors you never detected. A medical clinic's internal system records a code for the medical procedure performed, as well as the gender of the patient. It is discovered in data profiling that procedures are being recorded that are not possible for the gender code recorded. These are inaccuracy facts.*

However, the root cause is that the procedure codes are handwritten on paper forms and then sent to the data entry office. Many of them are illegible or missing. The data entry staff has no way of verifying the correct procedure and are motivated to get the data into the system rather than fix it. In addition to the procedure codes being invalid in the case of gender conflicts, there are probably many other procedure codes that are wrong. However, because they are valid procedure codes, they are not detected.

The remedy called for having the data entered directly online by the administrators of the doctors instead of transferring paper documents to a central data entry function. Because so many errors were noted, the new form displays a text description of the procedure when it is entered with a confirmation button. This helps the administrators confirm that they have entered the correct code.

Checks were put in for gender/procedure code conflicts, as well as other conflicts, such as invalid patient age/procedure code combinations. In addition, administrators were educated on the importance of correct procedure codes. Because of the better data entry procedures, the number of errors prevented not only included those that were detectable but many others that were not detectable through analysis.

An additional problem with metrics is that data quality assurance departments often believe that this is the end of their mission. They define their work product as the metrics. However, metrics do not define the source of problems nor the solutions. To improve data quality you need to follow through on getting improvements made. To hand the responsibility for this to other departments is a guarantee that the work items will sit low on priority lists of things to do and will not get done expeditiously. The data quality assurance department needs to track and drive the issues through to solution.

Metrics are not all bad. They are often a good shock factor for driving actions. When you give management a presentation that says the HR database records revealed 700 inaccurate values, this can raise eyebrows and produce a call for action. Knowing that you have 700 and that the real number is higher can be motivation enough.

Often a single fact is more shocking than statistical metrics. For example, telling management that a profiling exercise of the birth date of employees

revealed that the youngest employee in the company has not been born yet and that the oldest was born before the Civil War is far more effective than a metric at getting across the point that improvements are needed *now*. (I did not make this up; it was an actual output of a data profiling exercise.)

Issues

The real output of the fact collection phase is a set of issues that define problems that need to be solved. A single statistic can result in an issue. For example, 30% of the purchase order fields have no supplier ID number. Alternatively, several facts can be grouped into one issue. For example, the customer name and address data is severely flawed: 5% of name fields have invalid names, 15% of address fields are inaccurate or blank, 12% of city fields are blank, 5% of city fields are misspelled, and 12% of Zip codes are invalid or blank. This single issue rolls up several inaccuracy facts into a single issue that needs to be addressed. Addressing each inaccuracy fact is an inefficient use of time.

Issues need to be recorded in a database within an issues tracking system. Each issue needs a narrative description of the findings and facts that are the basis for the issue. It is important to identify the facts and the data source so that comparisons can be correctly made during the monitoring phase. The information needed for the data source is the identification of the database used, whether samples or the entire database were used, the date of the extraction, and any other information that will help others understand what you extracted the facts from. In tracking the issues, all meetings, presentations, and decisions need to be recorded along with dates and persons present.

5.2 Assessing Impact

Each issue that has been created needs to be studied to determine the impact it has already had or potentially may have on the corporation. Somewhere along the line someone will ask the "so what" question about an issue. It is important to justify development and disruptive efforts to deploy corrective actions. It is important to document the value returned to the corporation for the time and cost spent pursuing issues.

This needs to be updated from time to time. It is usually impossible to compute the costs and benefits up front. One approach is to look at the facts and theorize on possible impacts. A brainstorming session with data analysts, business analysts, and others may be helpful. This will lead to activities to prove that the impacts have already occurred. Because impacts have not occurred does not mean they will not in the future. As the issues are worked

through the entire process, additional information about impacts may become apparent. These need to be added to the impact section.

Impacts Already Happening

The impacts may not be obvious to anyone but may be very real. For example, an issue that states that suppliers exist in the supplier's database multiple times may lead to speculation that you are not getting large enough discounts for volumes purchased over a year. Investigation may uncover that this is true (one department orders under one supplier ID and another department uses a second supplier ID for the same supplier). You can easily compute the discount difference, the volume of purchases made, and the value lost to the corporation. The cost of this type of inaccuracy is totally hidden until the issue is identified and pursued.

Sometimes an issue is created from an outside-in investigation and the cost is already known. Tying the external cost to facts is part of issue definition. For example, the external manifestation might be that the accounts receivable department spends x amount of people time per month correcting wrong information on invoices. The facts are the number of blank or inaccurate values found during data profiling. The facts back up the assertion that invoices are not being prepared properly.

Further investigation may reveal that not only is time being wasted but that payments are being delayed by a certain amount for two reasons: one is the lag in time in getting invoices out, and the other is that invoices sent out without corrections get rejected by the purchasing company, causing yet further delays. In fact, there may be a group of invoices that are never collected due to data errors on the invoices. This is an example of a single visible cost leading to facts about inaccuracies, which lead to the discovery of more hidden costs.

One point to consider is that a significant accuracy problem on a data element may indicate a bigger quality problem. In the case of the missing supplier ID, it is clear that if 30% of the values are missing, there is a real possibility that the process is flawed and that the supplier ID is not available at the time the data is entered. It is unlikely that data entry staff are that bad at their jobs. It is also clear that this field is not involved in making the purchase or subsequent payments (it appears to cause no harm). The harm is all done in the secondary uses of the data. It is easy to speculate that if the data is not available at entry, data entry staff may also be entering wrong but valid values. The problem may be much larger than it first appears.

This is why you need to match inaccuracy facts to known manifestations. By seeing the actual data values in error and the data elements containing errors, you can often speculate about hidden costs that may be occurring.

Impacts Not Yet Happening

The most dangerous impacts are those that have not yet occurred. Seeing the presence of inaccurate data can sometimes lead to speculation about problems that could occur. These can have greater impact than those that occur on a regular basis but cost little to correct.

A simple example is the inaccurate birth dates of employees. There may have been no costs that have occurred yet for a new company that hires mostly young people. However, as this population ages, all sorts of government regulations about reporting, pension programs, and changing medical benefits when an employee reaches age 65 are at risk of occurring. These errors can also make decisions about hiring practices inaccurate and lead to wasteful efforts to adjust the company's mix of ages.

A business rule may require that a fast mode of shipment be used to ship certain materials that have the potential to spoil or decay. They may require refrigeration or avoidance of temperatures above a certain number. It may be that errors in the orders have caused a number of shipments to be made that violate the rule and no dire consequences have occurred. All values are valid individually, but the shipment mode rule for the product type is violated. By speculating on the potential for costs, the issues team may speculate about returned orders, merchandise that cannot be resold, and lost customers. However, that speculation may lead to the potential for real lawsuits, as the corporation may be liable for damage done to the purchaser trying to use spoiled merchandise.

This example may have been saving the company money (lower shipping costs) but creating a potential liability (lawsuits) that could severely damage or even destroy the company. This is why speculation on potential impacts is so important.

The process of assessing impacts will crystallize issues. It may result in issues being broken apart or issues being combined. As participants gain more experience, they will be better at sniffing out impacts both real and potential. As new participants join the process, they can benefit from the documentation of previous issues as a training device.

It should also be apparent that the documentation of the impacts of issues is highly sensitive information. The issues management process should provide for a high degree of privacy and safety of the information.

5.3 Investigating Causes

The next logical step in the process is to discover the causes of the inaccuracy facts. Remedies cannot be fabricated until more information is uncovered.

You need to perform a thorough study, in that the causes may not be what you think they are.

This chapter is not going to cover this topic comprehensively. This is a very large topic and beyond the scope of this book. However, a snapshot of some of the approaches is given to show the types of activities required.

Investigating causes requires talking to a lot of people in a lot of organizations. Assignments to investigators must to be done based on the substance of the issues. Participants from many organizations may be needed. The data quality assurance department should not try to undergo this step entirely with their own staff. Neither should they relegate this entirely to others. It is yet another place where the need for a larger team exists that gets guidance and leadership from the data quality assurance staff.

Investigation of the cause is not always possible. For example, databases purchased from vendors may be found to be defective. It is your responsibility to notify them of the problem and give them facts. It is their job to investigate the causes and correct them.

There are two basic approaches to investigating errors: error cluster analysis and data events analysis. The first is used to narrow down the sources of errors. The second is used to study the events that cause data to be created and maintained in order to help identify the root causes of problems. They can often be used together to efficiently complete the task.

Error Clustering Analysis

This type of analysis attempts to use information in the database to provide clues as to where the inaccuracies may be coming from. It starts with information about the specific database objects containing inaccuracies. For example, in an order database, it would start by identifying those orders that contain inaccurate data or that are suspected of having inaccurate data. Although many rules about data cannot identify specific data elements that are wrong, they can identify entire orders that contain the wrong data. The collection of all orders that have wrong values or rule violations constitutes the analysis set.

The analysis set may be defined narrowly (all orders violating a single rule) or broadly (all orders violating any rule). It depends on the amount of data in the analysis set and the importance of the individual rule. There is also the concept of rules having affinity. That is, for example, all rules that deal with the initial capture of the order information (a process clustering) or all orders dealing with customer name and address information (data semantic clustering).

Once the set of data is isolated that contains offending data, all of the data elements of the isolated set are used to determine if they vary in significant ways with the general population of data.

Common data elements that may reveal significant variances are data source location (branch office, geographic region, specific sales reps), customer information (first-time customers, Internet customers), dates (specific dates, days of week, range of dates), product type or characteristics (engine pats, volatile, expensive), or process steps completed (initial entry, order shipped, invoice created). You are looking for any factor that may indicate a starting point in examining the causes of the errors. Performing error clustering analysis can shorten the search for causes significantly through performing a relatively quick and simple test of data.

Data Events Analysis

This involves a review of all processes that capture data or change data. Data takes a journey from inception to one or more databases. It may have a single process event (data entry) or a number of events. The points of examination can be any or all of the following:

- data capture processes
- durations in which data decay can occur
- points at which data is extracted and added to a different data store
- points at which data is converted to business information

DATA CAPTURE PROCESSES

The process point at which data is captured represents the single most important place data can be made accurate or inaccurate. All data capture points need to be identified and examined. Some data is only captured once. Some is captured and then updated on an exception basis. Some data is captured and the business object updated or enhanced through a work flow process that may occur over a long period of time. Some of these points may take on multiple forms. For example, an order may be entered by the actual customer over the Internet, entered by a recording clerk from a form received in the mail, or entered by a company sales representative through a company client server application. This example shows three very different and distinct ways of entering the same business object.

Building a diagram of the data paths of a business object, identifying the distinct points of data capture, and specifying the characteristics of each is a time-consuming but extremely important task. Figure 5.2 shows some of the characteristics that need to be identified for each data capture or update point. Comments on these factors follow:

- Time between event and recording
- Distance between event and recording
- Number of handoffs of information before recording
- Availability of all facts at recording
- Ability to verify information at recording
- Motivation of person doing recording
- Skill, training and experience of person doing recording
- Feedback provided to recorder
- Data value support of getting it right
- Auto-assist in recording process
- Error checking in recording process

FIGURE 5.2 Factors in evaluating data capture processes in the data capture environment.

- *Time between event and recording:* In general, the longer the time differences, the greater the chance for errors. If the time lag is long enough, it also lends itself to missing or late information. Examples of long durations are cases in which forms are completed and mailed to a data entry location. The accuracy and timeliness would be enhanced if the time difference were eliminated through a more direct entry, such as through the Internet.

- *Distance between event and recording:* Physical distance can also be a factor. This reduces the opportunity for the person who is entering the data to verify or challenge information. For example, if the originator of data is in Chicago but the information is transmitted via telephone or paper to Kansas City for entry, you have a distance between the person who knows the right information and the one entering it. If there is confusion, the entry person has to either enter nulls or enter a best guess.

- *Number of handoffs of information before recording:* The first person to experience the event is most likely to be the one with the most accurate description of the facts. Each handoff to another person introduces the possibility of misreading written information, misinterpreting some else's comments, or not knowing information that was not passed on.

- *Availability of all facts at recording:* If the person entering the information has no access to the event, to the person who created or observed the event, or to databases containing important auxiliary information, they cannot fill in missing information or challenge information they see. For example, it is better for HR data to be entered with the employee sitting next to the entry person, as opposed to copying information from a form. Another example is to have a search function for customer identifiers available for order entry personnel.

- *Ability to verify information at recording:* This is similar to the previous issue, but slightly different. Can the data entry person get to correct information if they think the information provided is wrong? An HR data entry person could call or e-mail the employee if there is confusion. Sometimes the process makes it impossible to make this connection. Sometimes the process penalizes the data entry person for taking the time to verify questionable information. All entry points should allow for information to be either verified immediately or posted to a deferred process queue for later verification and correction if needed.

- *Motivation of person doing recording:* This is a complex topic with many sides. Are they motivated to enter correct information? Are they motivated and empowered to challenge questionable information? Are they motivated to enter the information at all? Someone entering their own order is motivated to do it and get it right. Someone entering piles of form information they do not understand could not care less if the information is entered correctly or completely. Is feedback provided? Is their performance measured relative to completeness and accuracy?

- *Skill, training, and experience of person doing recording:* People who enter the same information for a living get to learn the application, the typical content, and the data entry processes. They can be trained to do it right and to look for red flags. People who enter data on a form only one time in their life are much more likely to get it wrong. Sometimes there exists a data entry position that has not been trained in the application. This is an invitation for mistakes. Note that entry people who are making mistakes tend to make them repetitively, thus increasing the database inaccuracy level and thereby increasing the likelihood that it will be exposed through data profiling analysis.

- *Feedback provided to recorder:* Feedback is always a good thing. And yet, our information systems rarely provide feedback to the most important people in the data path: those entering the data. Relevant information, such as errors found in computer checks, should be collected and provided to help them improve the accuracy of data they enter.

- *Auto-assist in recording process:* Do the data entry programs and screens help in getting it right? A complex process can include pull-downs, file checking, suggestions on names, addresses, questioning of unusual options or entry information, and so on. Remembering information from the last transaction for that source can be very helpful in getting information right. Letting each data entry station set its own pull-down defaults can reduce errors. Providing the current date instead of asking that it be

entered can improve accuracy. There are a lot of technology best practices that can improve the accuracy of information.

- *Error checking in recording process:* Evaluate the checking provided by the entry screen programs, the transaction path, and the database acceptance routines. Data checkers, filters, and database structural enforcement options can all be used to catch mistakes at the entry point. These are not always easy to identify because they require someone to dig around in code and database definitions. Many times these are not documented. Many times they are thought to be true but have been turned off by a database administrator to improve performance. Many times they exist but are not applied to all points of entry.

It is important to study all factors at each entry point, even though the investigation started by focusing on a single set of inaccuracy facts. This process may reveal other inaccuracies that were hidden from the profiling process or uncover the potential for problems that have not yet occurred. It may also uncover some locally devised practices that are good ideas and may warrant propagation as a formal methodology throughout the data entry community.

DATA DECAY

The analyst needs to identify data elements that are subject to decay and check for process steps that exist that will mitigate decay. Identifying data decay candidates is a business analyst topic best handled as work sessions with participants from multiple departments.

If the investigation reveals that no procedures are present to prevent decay, the analyst needs to determine the extent to which decay has contributed to currently visible problems or whether it presents the potential for future problems.

Decay problems are often not observable though data profiling because the values in the database are valid even though wrong. However, process analysis may suggest that the data is susceptible to decay problems. Sampling the data and testing it through object reverification may reveal hidden problems. These can become the subject of new issues split off from those that got you there.

DATA MOVEMENT AND RESTRUCTURING PROCESSES

Many errors can be introduced when data is extracted, reformatted, aggregated, and combined with other data. If the data source that was used for identifying the inaccurate data is not a primary data source, it requires examination of the processes that build that database from the primary sources.

The first question to ask is whether the problems also exist in the original data source, are part of the data movement processes, or are the result of an incompatibility with the target database structure or definition. Errors at this level often cause primary data sources to be blamed for problems not of their making.

One of the problems with this type of analysis is that the extraction, transformation, cleansing, and loading processes are often not well documented or are documented only in the proprietary repositories of individual products used for the separate steps. This requires expertise on each of these repositories and on the functions of the individual products used. This can lengthen the time required to perform the analysis.

Often data movement processes are locally developed without the aid of packaged tool software. The project team merely writes code for each step. In these cases, finding out what the team does may be difficult because much of it is probably not documented at all. This stresses the importance of being disciplined enough to create and maintain metadata repositories on all data structures: primary, intermediate, and summary. Information should also be kept on all processes that move data between them.

Review of upstream processes may be indicated by discovering information about quality problems in primary databases. This means that a situation discovered in a primary database that produces inaccurate data may lead to the discovery that upstream uses of this data are also flawed. You are basically asking the question "What is the data warehouse doing with this wrong stuff?" This process of examining known data flaws through to their final use can raise issues that were otherwise hidden.

CONVERSION TO INFORMATION PRODUCTS

Other places to look are the conversion of data from databases to reports, movement to OLAP cubes, staging data in corporate portals, and other business information products.

This type of review would normally only be done if the issue were created from concerns raised about these objects. Looking at wrong output does not always indicate that the data is wrong. The routines to extract the data and to compute from it, and the timeliness of this activity, can lead to inaccurate business information products from perfectly accurate data. Problems in the information products should be traced back through the system because they can often uncover previously hidden problems with other uses of the same data.

It should be clear that the process of identifying where errors creep into databases has many beneficial side effects. It can surface bad practices that are creating errors that were not detected in the initial analysis. It can detect bad

practices that are not generating errors but have the potential for doing so. It can identify hidden problems in upstream copies of the data or uses of the data that were not known. This may lead to expanding the impacts section to include impacts already occurring and those that have not yet occurred. This process may lead to the consolidation of issues (discovery that the data entry process caused many of the issues) or creating new issues (the corporate portal is displaying flawed renditions of the data).

It may be helpful to document the bad practices independently for the benefit of future projects. Bad practices used in one application frequently find their way into other applications. The same team that implemented them in one case may have implemented them in other applications they also worked on. Having a list of bad practices can serve as a checklist of things to look for in subsequent investigations.

5.4 Developing Remedies

Remedies to quality problems can range anywhere from simply holding a training class for data entry personnel to replacing an entire application. Without remedies, the problems are likely to persist, if not get worse. Without remedies, the potential problems that have not yet occurred increase in likelihood of occurring.

Often the problems that exist in a database cannot be repaired. This is true when the number of errors make it impractical to seek out and repair the wrong ones. This is also true when it is no longer possible to obtain the correct information. The remedies are mostly designed to improve the quality of *new* data being entered into the databases as opposed to fixing the data that is already there.

There are a number of classical problems associated with this phase. The first is the trade-off of making quick improvements through patching an existing system versus taking a longer-term view of reengineering the data processes and application programs. The second is the trade-off between making changes to primary systems versus performing data cleansing to fix problems when moving data. Figure 5.3 lists some of the types of remedies that can be used for resolving issues.

Scope of Remedies

Remedies are changes to systems that are designed to *prevent* data inaccuracies from occurring in the future, as well as to *detect* as many of them as possible when they do occur. The scope of changes includes data capture processes,

- Improve data capture
  - Train entry staff
  - Replace entry processes
  - Provide meaningful feedback
  - Change motivations to encourage quality
- Add defensive checkers
  - Data entry screens
  - Transaction servers
  - DBMS implementations
- Add periodic monitoring
- Perform periodic data profiling
- Use data cleansing
- Reengineer and reimplement application
- Reengineer and reimplement data extraction and movement
- Educate user community

FIGURE 5.3 Data quality issue remedy types.

primary applications that create and update data, processes that move data between databases, and applications that generate information products. In short, everything is fair game to designing remedies.

DATA CAPTURE PROCESSES

Improving data capture processes can include actions such as redesigning data entry windows and associated logic, training data entry people, and instituting feedback reporting of quality problems to data entry people. Many small items like these can make large improvements in the accuracy of data.

At the other extreme is altering the business processes that include data capture and update. Changes in who enters data and when they do it can improve the efficiency of the processes and the likelihood that the data will be accurate. Getting the entry of data closer to the real-world event, having fewer people involved in the process, and having the entry people trained on the intent of the application can all contribute to better data.

Business processes can be altered to add data verification through additional means in cases where it is warranted. Business processes can be altered to eliminate incentives to provide inaccurate data.

More automation can be brought to the entry process wherever appropriate. Use of bar coding, lookup of previously entered information, voice capture of verbal information exchange between the person creating the data and the person entering the data for later replay, and verification are examples where automation can improve accuracy.

ADDING DEFENSIVE CHECKERS

Defensive data checkers are software that assists in enforcing rules at the point of data entry to prevent invalid values, invalid combinations of valid values, and structural problems from getting into the database in the first place.

Rule checking can be performed in multiple places and through multiple means. Data entry screens can be designed to check for valid values for encoded fields and to enforce values for required fields. Application server code can take the data for a transaction and perform further rule testing for more stringent value testing and multivalued correlation testing. The database implementation can employ the support of the DBMS software to enforce many structural rules, such as primary key uniqueness, primary/foreign key constraints, and null rule enforcement. The use of a separate rule-checking component can be added to the transaction flow to perform additional data rule checking.

A solution that might be chosen is to leave the application alone but change the database management system used in order to take advantage of a different DBMS's superior data-checking functions.

Data checkers can be moved more into the mainstream of the application. For example, several new Internet applications are checking the correlation of address information at the point of data capture and alerting the entry person when the various components are incompatible.

Defensive checkers cannot prevent all inaccuracies from getting into the database. Inaccuracies still flow through in cases for which values are valid individually and in combination but are just plain wrong. It is also generally impractical to test rules that involve large sets of data to determine correlation correctness.

ADDING DATA MONITORING

Data monitoring is the addition of programs that run periodically over the databases to check for the conformance to rules that are not practical to execute at the transaction level. They can be used to off-load work from transaction checks when the performance of transactions is adversely affected by too much checking. Because you can check for more rules, they can be helpful in spotting new problems in the data that did not occur before.

DATA CLEANSING

The use of data cleansing programs to identify and clean up data after it has been captured can also be a remedy. Cleansing data is often used between primary databases and derivative databases that have less tolerance for inaccuracies. They can also be used for cleaning up data in original source systems.

Data cleansing has been specifically useful for cleaning up name and address information. These types of fields tend to have the highest error rate at capture and the highest decay rates, but also are the easiest to detect inaccuracies within and the easiest to correct programmatically.

REENGINEERING APPLICATIONS

In extreme cases, the application that generates data can be overhauled or replaced. This is becoming more common as the solution to cases in which many data issues pile on the same data source.

Reengineering can apply to the primary databases where data is initially captured, as well as to the applications that extract, transform, and move the data to derivative data stores or to the derivative stores themselves.

This remedy rarely stands alone. All other remedies are specifically directed at solving a data quality problem. Reengineering generally will not be selected as a solution solely for data quality reasons. Data quality concerns become additional justification for making a change that has been justified by other business drivers.

Short-Term Versus Long-Term Solutions

Remedies need to be devised with consideration for the cost and time to implement. Time to implement must include the time lag before it is likely any project would start. Many of these remedies require negotiation with development teams and scheduling against many other competing tasks.

This often leads to a staged approach to implementation involving data cleansing and monitoring early and reengineering of applications later. It may also lead to implementation of throwaway efforts in order to effect some short-term improvements while waiting for long-term projects to complete.

Too often projects initiated from these remedies end up on a to-do list and then get dropped or continue to get prioritized behind other projects. A reason for this is that they tend to be too granular and are not competitive against bigger projects that promise greater returns.

Issues management should strive for as many easy or short-term remedies as possible to obtain quick improvements. For example, training data entry people, changing screen designs, adding checker logic, or setting expectations are easy to do.

Data cleansing can also be introduced as a short-term remedy to fill the void while more substantive changes are made. Data cleansing should always be considered a temporary fix.

These are tricky matters to manage. One of the dangers is that the temporary improvements become permanent. Managers think that because some improvements have been made that the problem is solved. They may think that data cleansing is a solution instead of a short-term coping mechanism.

This underlines the need to keep issues open as long as they are not fully addressed. If necessary, long-term remedies can be split off into separate issues for tracking.

This is also a reason to monitor the results of remedies implemented. After the short-term remedies are implemented, the profiling process should be repeated and the impacts reexamined. This allows quality problems and their associated impacts that remain after short-term remedies are implemented to be documented, sized, and used to justify the longer-term efforts.

Practical Versus Impractical Solutions

There is a real danger in this phase of overengineering remedies. A zealous data quality team can outline a number of measures that will have no chance of being implemented. It is important that the team performing the remedy recommendations include representatives from the IT and user organizations in order to avoid recommending something that will be rejected.

An example of overengineering is to require that all data rules discovered during the data profiling process be implemented as transaction checks or as periodic monitoring functions. Although this would catch many errors, in practice it has the potential of overloading the transaction path and causing performance problems. The rule set needs to be prioritized based on the probability of errors occurring and the importance of an inaccurate value. The high-risk rules should be added to the transaction path, moderate-risk rules should be added to periodic monitoring sweeps over the data, and low-risk rules should not be implemented. Periodic reprofiling of data may check the rules not implemented to make sure they are not becoming more of a problem; possibly once a year.

Note that a rule can be classified as high risk even though profiling indicates few if any violations have occurred. If the potential cost to the corporation of a violation is very high, it needs to be included in checkers even though there is no evidence it has already produced inaccurate data.

Another example is to call for a major company reorganization to obtain more reliable data capture processes. This should not be considered a remedy unless an awful lot of evidence exists to justify it.

Organizations resist change, and change does not always produce the expected results. If there is a perception that little is to be gained, this type of recommendation will never be approved.

Similarly, recommendations that require major changes to high-availability applications are less likely to get approved. The disruption factor on a major application can cost a company tons of money if it is not managed properly. These types of changes are not accepted easily.

Turning Remedies into Best Practices

As more and more issues pass through the process, the team will learn more about what types of remedies are most effective and what types of remedies can more easily be adopted. What you learn can be converted into best practices that can be employed in all new system developments. This is a good way to improve the quality of data coming from new systems before a data quality problem even exists.

This is a part of the role of the data quality assurance department. It feeds into their role of *preventing* problems.

5.5 Implementing Remedies

Implementing remedies almost always involves other departments, their budgets, and their staff. The quality team needs to present very solid cases for improvements.

Quality problems are identified and impacts assessed at a granular level. Trying to get implementation commitments on individual issues generates a great deal of resistance. Issues need to be combined into change initiatives that have a bigger impact on the value returned to the corporation. This also leads to efficiencies in implementation.

The data quality assurance team needs to monitor implementation because items often get accepted for implementation but never get done or get done improperly. This is another reason for periodic reviews of all open issues. You need to keep the problems highly visible until they are no longer problems.

5.6 Post-implementation Monitoring

It is important to continue monitoring databases after remedies have been implemented. This provides two distinct values: validating the improvement effort and checking for the occurrence of new problems.

Validating Changes

The need to measure the quality of data before and after changes accomplishes two things: it validates that the changes have had a positive impact, and it quantifies the value provided to the business. The next chapter covers the factors considered in justifying data quality assurance functions. It demonstrates that it is impossible to predict the effects in advance. The best indicator of the potential value of an investigation is the value returned from other, similar investigations. This demonstrates the need to measure again after changes.

Also remember that the real impact will never be known for sure. If an inaccuracy count of 5% was found before changes and only 0.5% after changes, a logical conclusion is that an impact has been made. However, the real but unknown rate before may have been 7%, and after, 1%. There is a sizeable impact, although the true statistical difference is not known. This underscores the need to keep metrics from being used as absolutes but rather as indicators of the direction and relative size of impacts.

If post-change monitoring does not show significant differences, the analysis of causes or the implementation of remedies was not successful. The issues team needs to circle back and rethink what they have done.

Sometimes changes have an unintentional negative impact. For example, performance may be severely degraded due to extra error checking, or the number of rejected transactions may become too high. The trade-off is between "Do I let incomplete and inaccurate information get through in order to get the transactions processed?" or "Do I insist on perfectly accurate information before any transaction can complete?". There is no need to compromise quality to obtain accurate information, although it may take a lot of work and innovative process design to achieve it. The first attempts may not prove to be the optimal solution, and additional attempts need to be made.

Impacts on the business process should also be observed and documented in the issue management system. These may be positive or negative. Often streamlining the business processes to obtain higher-quality data leads to other savings as well.

Continuous Checking

All information systems should be instrumented to provide ongoing monitoring of data quality parameters. Most older systems have little or no monitoring functions built into them. They should be retrofitted into systems when addressing important quality issues. They should be included when developing new applications or making major renovations to existing applications.

Monitoring can include a number of things: feedback on rejected transactions, periodic execution of rule sets over databases, and periodic thorough data profiling.

Feedback on rejected transactions is important because excessive rejection indicates poor business process design. It is easy to accomplish this, but it is rarely done. Indications of the quantity of rejects, the point of rejection, and the data elements causing the rejection provide valuable information to data quality assurance staff, application developers, and business process designers.

An example of this is to design an Internet order form such that every time the user has a SUBMIT function denied because of checking errors a quality packet is built and sent to the application server indicating the errors found. The alternative is to wait for a correct form completion and only send that. The last approach provides no feedback that could lead to better form design and less frustrated customers.

Continuous monitoring tends to become decoupled with issues tracking. This is because the monitoring mechanisms become more global in nature and take in monitoring of information relevant to many issues. At the least, the issue tracking system should identify the specific additions to monitoring functions performed as a result of that issue.

5.7 Closing Remarks

This has been a light trip through the process of developing, solving, and measuring the effectiveness of issues that come from data quality processes. The emphasis has been on how issues are created and managed that originate from data inaccuracy discoveries.

This treatment should cement the thought that data quality improvements are long-term and very public tasks. The data quality assurance group cannot function in isolation. The other departments engaged in the data acquisition, management, and use activities are very integral parts of the process and need to be included in the process at all steps. They also need to accept the goal of better-quality data and to welcome efforts rather than resist them.

Issues can have a very long life. I suspect that some of them can live forever. This leads to the need for formal treatment of them as business objects. It also calls for issues to be very accessible in their own database.

Issue resolutions are often considered interruptive to the normal flow of work through departments that develop and deploy information technology. They will tend to get sidetracked easily if not monitored and placed in front of management on a regular basis.

These activities need to become the normal flow of work. Monitoring data quality and making corrections to improve it should not be considered a nuisance, but should be considered a regular part of information systems operations. This chapter again highlights the need to coordinate the activities of data quality assurance with the complete information systems agenda.

I AM *not aware of any issues management software that has been developed specifically for data quality issues. The best available software is standard project management software, of which there are many flavors available. Most organizations are already using one or more of these packages. It would be helpful if some vendor addressed this topic specifically as it relates to data quality assurance.*

Issues management would make an excellent XML database application. The different phases and types of information are easy to generate standard tags for. Access to information over the Internet would be facilitated through this approach. This is important, considering the wide range of people who need to get involved in issues at one point or another.
