



"turning data into dollars"

Tom's Ten Data Tips – July 2009

Data Quality Policy

There's a direct line from Corporate Governance to IT Governance to Data Governance. Since the Sarbanes-Oxley act (SOX) and similar regulations in Europe, CEO's are more aware than ever of the risks involved with erroneous (financial) reporting. Data quality policies ensure compliance with reporting obligations, and send a message of "care" to customers.

Data quality can affect customer transactions, work involved in reporting, meeting with regulatory obligations, compliance, etc. Customer transactions can go awry because contact details were misspelled, or the contact history wasn't captured properly. This will result (mainly) in reputation risk. Reporting can become time consuming and error prone because of inconsistent definitions, inaccurate, missing or late data, and a raft of other quality issues. All of these negatively impact the bottom line.

1. A New System Is *Not* The Solution To Data Quality Problems

The source of data quality problems is often attributed to "systems." And in some way or the other systems *are* always involved. Careful design *can* enable making quality the default, but the root causes for problems with data quality lie *never* with systems *alone*. Instead, working practices, management objectives and accountabilities lie at the heart of these problems.

Since "systems" are easy to blame, never argue, yet are tangible, they are an obvious scapegoat. And on top of that, installing a *new* system is the kind of decisive management action that engenders trust. This is why "a new system" is such a convenient way to externalize the problem(s). But if it's the same people that will use the new system, in similar roles and with similar responsibilities, think twice about how viable this can be as a sustainable, long-term solution.

2. Data Stewards Can Not Resolve Data Quality Problems (But They *Are* Important)

One of the latest "best practices" that Gartner has advocated is appointing a designated data steward. In principle there's nothing

wrong with this, *per se*. A more interesting question is when and how this role will add sufficient value to offset the cost. Has an existing staff member been “promoted” to data steward? Then what about his existing duties? Although a laudable job, care must be taken how it is launched within the department.

For this role of data steward to be effective, he needs direct access to a (very) senior manager. However, if his effectiveness relies solely on the senior executive’s clout, on playing his “trump card” too often, chances are that the effects will erode fast. So he needs this indirect influence, but if everyday practice requires him to actually use that threat more than incidentally, this won’t work. It should be clear to all involved that the senior sponsor expects *not* to be involved in settling disputes very often, and that you’d better have a very good case if you want to bring something up.

3. Good Data Entry Forms Mitigate Problems

Manual data entry is *always* a cause for data quality concern. The number of errors you may expect range roughly from 1%-20% (typically around 5%), depending largely on the ergonomics of the application, awareness about costs for non-quality and quality control (and feedback!) in the back-office. We have observed repeatedly that a *fast* (convenient, user-friendly) application is usually a *high quality* one.

Technical accelerators allow operators to work more accurately as well as quickly. This may involve visually and logically organized fields, making text mainly lower case for readability, making it clear which fields are optional, avoiding “look-up” procedures which take staff away from the screen, providing user-friendly field controls, etc.

In particular high cardinality (many distinct values) nominal fields like for instance “occupation” are a challenge. Several spellings and/or descriptions need to map onto one class. “Smart” lookup like the search aids one knows in Google help operators “search” while they type in text, and then offer an ever decreasing set of categories to choose from. Not easy to design, but necessary to get a reasonable level of quality, while supporting fast data entry.

4. Data Quality And Meta Data Go Hand In Hand

Data integration often surfaces inconsistencies in data definitions and/or formats. Of course in theory these should be addressed by the

meta data at hand. Reality isn't always perfect, however. Issues can be something 'innocent' like different field lengths for the same attribute, or more pernicious like "overloaded" fields. Non-matching field lengths (for instance name or address) cause problems when some of the records have been truncated – the logic for this needs to be surfaced (from the meta data) to get a good match. Overloaded fields are both very bad practice and scarily common.

An "overloaded field" is used for more than one purpose. For instance, when the "balance" column contains 99.999, = this means the branch manager has closed a special deal with the customer, and the details for this are to be found in yet another column. Overloading fields is bad enough as it is, but to make matters worse, they are often poorly documented. IT staff often find out about this "the hard way"...

5. 'Temporarily' Lifting Attribute Value Constraints Is *Not* A Good Idea

Sometimes for purposes of data entry, or when migrating external data sources, attribute value constraints are 'temporarily' lifted. In the case of data entry this is done to enter a value *outside* the valid range. This implies that the data entry application doesn't properly support the primary workflow because the input rules are too restrictive. In the case of a source migration, there is often insufficient time to either profile the source data and/or redesign the parameters of the target database. As a result invalid data enters for 'future' reconsideration. And then "future" has this nasty habit of being some inconvenient occasion...

When data won't quite "fit" during a data conversion project, it is not going to "fit" afterwards, either. While time pressure is the argument to lift the attribute value constraints, it is actually more time consuming to fix problems afterwards. Sometimes much more so, to the point where the erroneous data become 'native' to the new expanded database.

6. Choose "Valid Value Ranges" Based On Misclassification Costs

Usually there is no exact formula to determine the "valid value range" for a variable. Take age, for example. It is clear that ages over 150 can hardly be valid entries. But where you would choose to discard entries as untrustworthy is impossible to say with absolute certainty.

Attribute value constraints need to be chosen with both type I and type II errors in mind.

When the valid range for age is set from 0-150 it is clear that all “real world” values can be entered without a problem. It is questionable, however, how many (*if any*) of the entered values “149” actually represent people of that age. The real world value is more likely to have been “14”, “19”, or “49” with an additional digit entered erroneously. If the valid value range were set from 0-99, people over age 100 would pose a problem to the system. Both types of errors and associated costs need to be considered when setting the valid range.

7. Balance “Free Form” Data Entry With Restricted Editing

Attribute value constraints define the sets or range of admissible values for a variable. An example of a set might be M, F, or U for Male, Female, or Unknown. An example of a range could be that the variable age must lie within 0-120. When these attribute value constraints are embedded in the technical metadata, only values within this range can be entered. Input rules that are too restrictive may produce missing or erroneous information if data entry clerks arbitrarily change values to fit into the field so as to pass an edit check. Sometimes they can't make a value “fit” and therefore need to skip this field. And back office employees *will* become knowledgeable and creative with the system!

8. Data Quality Improvement Leads To The Discovery Of New Data Quality Problems

Be prepared when you take off on a journey of continuous improvement that “perfect quality” will never be attained. As surface issues are resolved, new and deeper issues *will* emerge. Interestingly, moving from 96%-98%-99%-99.5% data quality tends to require about the same effort as you slash the number of defects in half. What is more important is that quality checks and working practices you install provide a stable equilibrium. Robust processes do not break down in the face of (unexpected) setbacks in quality.

9. Track And Analyze Data Quality Errors

On the developmental path from errors “happening to you”, to control, to optimization, at some point you cannot get around an error tracking system. Much like helpdesk issue tracking, you want to trace evolution over time, ability to manage open issues, and some analytic capabilities. Are you successful in driving down issue-to-resolution times? Where are errors occurring? How often? How is that shifting over time?

Control and optimization (CMMI levels 4 and 5) absolutely require a thorough insight in what is happening, common error patterns, and associated costs for the business. At the operational level, you need to manage the workflow of errors, and your system should enable that as well.

10. Embrace Data Quality Policies In Your Governance

Poor quality data is rarely an isolated “event.” It is often related to lack of meta data, no central data definitions, immature or lacking Master Data Management, etc. Your data quality policies are bound to work best when part of a broader data governance initiative.

Isolated efforts to improve data quality can be worthwhile, and might even show a positive business case. For enterprise wide data consolidation however, central data governance is required to ensure data quality efforts across the organization are aligned. An orchestra without a conductor cannot perform, not even with the world’s best musicians...