

The Shortcomings of Two Common IT Testing Solutions

Michael Matteson – May 25, 2010

Masking Data & Manually Created Data

Masking Data – Data Masking is the process of augmenting sensitive data so that the data is still somewhat representative of the source, without being sensitive. Common methods of data masking include [http://en.wikipedia.org/wiki/Data_masking]: encryption/decryption, masking (i.e. numbers letters), [substitution](#) (i.e. All female names = Julie), [nulling](#) (####) or [shuffling](#) (zip code 12345 = 53412). These practices are typically used in a testing environment where using live production data is prohibited due to security reasons. While the idea can be attractive to product managers (as they believe they have a vast wealth of data that can be masked for testing) with the belief that 1) their data is secure and 2) that developing this data is trivial. Below are some troublesome points regarding the reality of Data Masking:

- The masked data still originates from private data, and so is never 100% secure.
- Masking methods can make it difficult to test business rules requiring a higher level of data realism.
- Data masking changes your data's original statistical distributions and won't deliver any control over your resulting distributions.
- The actual practice of Data Masking often occurs on provisioned non-production or less secure environments, leaving the real security of the data being masked the sole responsibility of the user.
- Certain information is statistically more difficult to mask when business rules are set in place to react to certain conditions. The masking of this data could cause numerous unintended anomalies within the data bringing about impossible/irrelevant test situations.
- Although using "real" data (masked or not) has superficial appeal, such data contains unintended errors and contains no metadata useful in interpreting test results.

As an example from the health industry, with the *American Recovery and Reinvestment Act* (ARRA) funding of many Health IT projects, vendors are in need of data to test their solutions for an improved Health IT Infrastructure and in doing so they need to comply with the *Health Insurance Portability and Accountability Act* (HIPPA). To comply with the HIPPA Privacy Act, vendors need to use what is termed as a HIPPA *De-identified* data set. These are datasets derived from real patient data where the data is then manipulated as to remove the risk of re-identification. The government in 1996 outlined two ways in which this should be accomplished. The two methods are as follows:

1. The "safe-harbor" method, requires the removal of every one of 18 identifiers enumerated at section 164.514(b)(2) of the Privacy Rule. [See Table 1] Data that are stripped of these 18 identifiers are regarded as de-identified, unless the covered entity has actual knowledge that it would be possible to use the remaining information alone or in combination with other information to identify the subject.

The Shortcomings of Two Common IT Testing Solutions

Michael Matteson – May 25, 2010

2. To have a qualified statistician determine, using generally accepted statistical and scientific principles and methods, that the risk is very small that the information could be used, alone or in combination with other reasonably available information, by the anticipated recipient to identify the subject of the information. The qualified statistician must document the methods and results of the analysis that justify such a determination.

Table 1: HIPAA's 18 Identifiers

- a. Names;
- b. All geographic subdivisions smaller than a State, including street address, city, county, precinct, zip code, and their equivalent geocodes, except for the initial three digits of a zip code if, according to the current publicly available data from the Bureau of the Census:
 - i. The geographic unit formed by combining all zip codes with the same three initial digits contains more than 20,000 people; and
 - ii. The initial three digits of a zip code for all such geographic units containing 20,000 or fewer people is changed to 000.
- c. All elements of dates (except year) for dates directly related to an individual, including birth date, admission date, discharge date, date of death; and all ages over 89 and all elements of dates (including year) indicative of such age, except that such ages and elements may be aggregated into a single category of age 90 or older;
- d. Telephone numbers;
- e. Fax numbers;
- f. Electronic mail addresses;
- g. Social security numbers;
- h. Medical record numbers;
- i. Health plan beneficiary numbers;
- j. Account numbers;
- k. Certificate/license numbers;
- l. Vehicle identifiers and serial numbers, including license plate numbers;
- m. Device identifiers and serial numbers;
- n. Web Universal Resource Locators (URLs);
- o. Internet Protocol (IP) address numbers;
- p. Biometric identifiers, including finger and voice prints;
- q. Full face photographic images and any comparable images; and
- r. Any other unique identifying number, characteristic, or code

Taken from: <http://irb.jhmi.edu/HIPAA/deidentifieddata.html>

Using method 1 can do so much damage to the original data's integrity that it is useless for sophisticated performance testing. If a clinical decision system needs to reroute or disposition a case based upon repeat offenders Social Security numbers, you would not be able to test that business rule with HIPAA de-identified data set.

Method 2 is rather vague, but also extremely difficult to do. Consider an area containing over 20000 people, which means the first three numbers of the zip are known (see Identifier stipulation b.-->i.).

The Shortcomings of Two Common IT Testing Solutions

Michael Matteson – May 25, 2010

Now consider in the de-identified data set there exists several rare diseases. This can be cross correlated with social web groups, medical portals, and IP information stored at these public health records or social group sites to pinpoint location and thusly the subjects identity.

Manually Created Data – Many developers just attempt to manually create data sets for testing. This is slow, costly, and produces low quality data. Typically these data sets are built manually by engineers costing valuable development time and money. Because humans are involved in this complex process, all the needed scenarios are not thoroughly tested leaving unknowns with regards to system or sub-system performance and business rules that are not exercised.

The required volume and complexity of data needed for effective performance testing makes it practically impossible to manually create good quality at any cost!