Business Applications: On the Tension between Efficient Testing and Compliance

Klaus Haller, Swisscom Enterprise Customers, Pfingstweidstr. 51, CH-8004 Zürich

1. Introduction

Smooth business processes need a stable IT landscape. Thus, IT departments spend time and money on testing their business applications. But what could be a reason for a tension between efficient testing and compliance needs?

Sensitive data are the reason! Testing business applications requires adequate data in the databases of test systems. Such data are often sensitive: client data in banks, patient records in clinics, patents to be filed, etc. If production data is copied to test systems, many testers and developers have access to sensitive data. Also, this can imply transferring sensitive data to other jurisdictions or to outsourcing partners. This is not only a risk. It might violate laws. This paper helps testers staying compliant with regulations without excessive costs. It is based on previous work on test data [1,2] and information security and data loss prevention [3,4,5].

2. Test Data Management and Efficiency

When testing a calculator, the correct result of a test case depends only on its input values. "3" "+" "4" and "=" is always "7." In contrast, the outcome for many test cases for business applications also depends on database data. The account balance in the database decides whether a bank client can withdraw 50€ with his debit card. This test case, however, can only be executed if the core banking system can start up and reaches the starting point of the test case, so hundreds of database tables must be populated with data. Thus, a test case for business applications must state what to type in into the GUI (e.g., 50€) plus the *database system state* to get to the starting point of the test case and the *database test objects* (e.g. accounts) needed for a concrete test case (Figure 1).

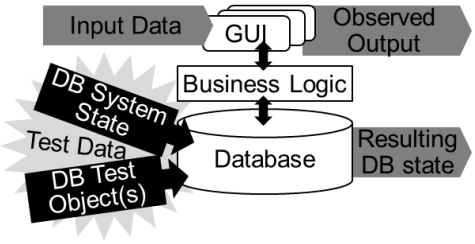

Figure 1: Business Applications and Test Data

Test data management looks on processes, test center organization, and tools in two areas: providing test data (creating or identifying test data in databases based on clearly defined requirements) and managing test data types. The latter is crucial, but often overlooked. Test centers must manage *test data types* or they lose their investments into test cases. Test cases are only repeatable for years if the data requirement is clear. A test case must not state "test with adequate account." Better, but only slightly, is "test with account 1234567." Mostly, the life span of test cases is longer than the life span of data in the database. Accounts can be closed or modified. Then, test cases fail and are "lost." Thus, test cases must state the *type* of data needed, e.g., "account with 370€" (see [1] for details).

Test data management has to be reflected in the test center organization. If test data types are defined, test centers can centralize the test data provisioning, which can save costs or improve quality. In contrast, test data type management must be enforced centrally, but remains the work of all the test analysts writing test cases.

3. Towards "Clean" Test Environments

3.1 Reduced-Sensitivity Environments

A cost-effective and easy way to get good test data is copying production data to test systems. This remains the standard for non-sensitive data. When it comes to copying sensitive data to test systems, more and more concerns are raised by risk or legal and compliance departments [5]. In the following, we present five options to address risk and compliance objections (Table 1).

Basic database masking addresses the risk of losing bulk data. Sensitive attributes are masked during the copy from production to test: real names are replaced with random ones, letters and digits in free text fields are replaced with "X." etc. When applied to all data copied to test systems, testers, for example, cannot extract and sell illegally lists with all clients, offers, patents, etc.

A bank can focus on preventing that complete customer lists are stolen. Instead, and more strict, the bank could state to the IT department: Please ensure that testers cannot identify *single* clients by looking at test data. In the first case, basic data masking is sufficient, but not in the second case. Preventing bulk data loss is easier than making it impossible to reconstruct production data based on masked test data. In the latter case, more attributes are a risk. Often, for example, testers can identify a commercial client by knowing the ZIP code and that he is, for example, a butcher. This can be an issue for outsourcing or offshoring. Two countermeasures exist. During the copy process, the option *castrate & inject* removes all sensitive data plus all data useful for reconstructing sensitive data (e.g., industry sectors). Then, synthetic data are added so that data is available for most test cases. The option of *pure synthetic data* goes one step further. It does not copy any data from production to test systems. All test data is synthetic.

Table 1: Overview Approaches for "Clean" Test Environments

Option	Risk Mitigated	Scope	Copy Control & Monitoring
Copy production data to test	None	n/a	Not needed
Basic database masking	Bulk data existence	Database or complete environment	Needed
Castrate & Inject	Single data items recreation		
Pure synthetic data			
GUI level masking	GUI access to sensitive data	Application or role in an application	Not needed
On top test data			

Most test centers prefer the simpler option for cost reasons. As a rough estimate, costs raise by one magnitude when moving to the next complex option: from copy data to basic masking is one magnitude, another one for castrate & inject, and one more for fully synthetic data.

Two alternatives exist for systems tested by large numbers of testers working on the GUI level. Both copy production data to test database, but do not expose sensitive data to GUI level testers. In case of *GUI Level Masking*, the presentation layer masks sensitive attributes in test systems. The option *On Top Test Data* helps in case of multi-tenant systems. A dedicated test tenant with synthetic test data is set up. (Most) GUI level testers test using only this tenant.

3.2 Copy Control – Production-to-Test

The test center head must state clearly if test systems must not contain sensitive data. However, humans forget and they ignore rules. So it must be made impossible for testers to copy production data to test systems. The first step is to place test systems in a separate zone. Second, testers and developers must not be able to transfer data in and out of the testing zone. Finally, from time to time, data has to be transferred from production to test to analyze bugs. A dedicated team must be able to perform such copies. However, the team needs rules on when and how to copy data. Also, each team member must understand the sanctions for not following the rules.

3.3 Monitoring Test Environments for Sensitive Data

Creative testers find ways to load "dirty" data into testing systems. Uploading CSV files to test systems that have been exported from production is one way. Also, they could manually type in production data in test systems. If dirty data gets into test systems, it is like having cancer. First, it is an issue to be addressed itself. Second, the later cancer or dirty data are detected, the more metastases exist or the more other test systems are polluted. Thus, test systems must be scanned periodically for dirty data. SQL scripts work in smaller test environments. Larger test centers might prefer DLP tools [3], which support an industrialized approach based on clean-up workflows. They can search for sensitive data using pattern matching or search lists of data which must not be in test systems.

3.4 Where is the Business Case?

Test centers need funding for cleaning up their test environments. If there is a regulatory need, the IT department must provide the funding to the test center. If outsourcing or offshoring programs are the reason, they must provide funding. Certainly, this can impact their business cases.

4. Compliance Testing – New Tasks for Test Centers

Most test centers offer functional and performance tests, which ensure that end users can work with the tested systems. In recent years, new test needs such as security and compliance tests have become increasingly relevant. They are not always in the focus of development and test teams. However, failing in this area has severe consequences: bad press, loss of business secrets, regulator intervention, etc.

Mostly security experts perform security tests such as vulnerability scanning or penetration tests. When it comes to compliance tests, test centers can help to verify compliance with SOX, HIPAA, Data Privacy Acts, etc. They know how to test the existence or absence of features. In contrast to auditors, they can provide in-depth testing on an industrial scale. Sample test cases are:

- *Feature-related Test Cases* validate whether an application allows to search for highly sensitive data (e.g., for a client named "Angela Merkel") or has bulk export functionality for unmasked data (e.g., call records in the telecom sector).
- *Data-related Test Cases* look for sensitive attributes on GUIs. Free text fields and file attachments are particular critical for data privacy and protecting business secrets in cross-border and cross-company business processes.
- *Role Scope Test Cases* ensure that a role can see only data they need for their business processes. This is to detect if users see much more attributes just because the standard software does not provide fine granular access roles.
- *Compound role test cases* look at users with two or more roles. Do roles interfere resulting in "super powers"? The first role could see client lists, but not print or export them. A second role could search and print contracts of a single client. What happens if both roles are combined? Can complete lists be printed?

To conclude: Compliance can be a burden for testers and can make testing less effective. At the same time, new opportunities for testers emerge: managing test data or performing compliance tests. And if you are a manager who disliked rules coming from a compliance department, a wise quote from Criss Jami might make you feel better: "*I would rather be an artist than a leader. Ironically, a leader has to follow the rules."*

8. References

[1] K. Haller: *Test Data Management in Practice*, Software Quality Days 2013, Vienna

[2] K. Haller: *The Test Data Challenge for Database-Driven Applications*, DBTest'10, June 7, 2010, Indianapolis, IN

[3] K. Haller: *When Data Is a Risk: Data Loss Prevention Tools and Their Role within IT Departments*, ;login, February 2014

[4] K. Haller: *Data-Privacy Assessments for Application Landscapes: A Methodology*, WfSAC at BPM 11, Clermont-Ferrand, France, 2011

[5] K. Haller: *Testdaten als Risikofaktor*, SQ Magazin, December 2013

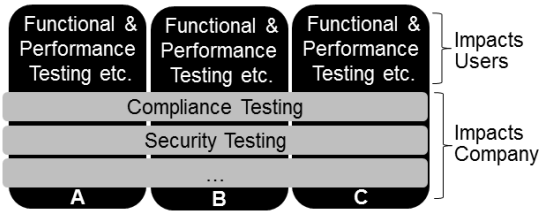

Figure 2: Compliance versus functional testing