

## Test methods in the TMS lifecycle

Testing in general – and especially in the context of systems as sensitive as those used in treasury management – is an indispensable aspect of both quality assurance and decision-making. The golden rule for all change processes is thus never to release system modifications or new developments for productive operation without adequate testing.

In this context, two key questions must be addressed: What test phases are necessary? And what constitutes necessary and reasonable cost given the need for meaningful findings?

Irrespective of whether changes are made as part of a project or within the framework of regular operation, several test phases are usually necessary. To begin with, functional tests verify whether the software delivers the guaranteed properties. It is also essential to test whether correct results are supplied. Integration testing – the next phase – examines interaction with other components in the system environment, as well as testing proper functionality and any restrictions in the context of the overall system. For treasury functions that can place a heavy load on the system (in payment transactions, for example), further tests to validate and assess system capacity and software behaviour under appropriate loads are also usually necessary. Here, distinctions can be drawn between load tests that push a system to its expected limits, stress tests that simulate a system overload, and robustness tests that examine system behaviour if individual components fail and under abnormal environmental conditions. In many cases, these phases are also complemented by security tests and/or migration tests.

The test phases normally conclude with acceptance testing in a system environment that simulates productive conditions as closely as possible. Formal acceptance is intended to ensure that the system satisfies the documented requirements. Ultimately, successful completion of this phase transfers all relevant risks from the project organization to the treasury department. This is also when the warranty phase begins.

Clearly, numerous test phases and correspondingly high costs can be a characteristic feature of this kind of project. However, the fact that such costs are generally justified becomes apparent when they are compared with the risks arising from system components that have not been properly tested – or indeed from complete system failures. High availability demands and strict demands with regard to critical treasury processes and throughput times are normally placed on treasury system environments, especially on modern, integrated environments that cover a broad spectrum of functions. It is therefore important to conduct a cost/benefit analysis and think very carefully about how much money can reasonably be invested to test changes to the overall system. To this end, models such as the constructive cost model (CoCoMo) can be applied to determine such factors as the cost, coverage and effectiveness of tests for each individual project.

In today's project organizations, various tests are often automated in order to optimize resource allocation. Whatever form tests take, however, it is vital to carefully design, log and document all tests to ensure that results, errors and deficiencies can be identified unambiguously and corrective measures can be taken without delay.

Lastly, testing should be an integral component of the training provided to users. In our experience, there is no better way to introduce users to new systems and functions and to familiarize them with systems than getting them to conduct their own tests.