Cash Management System and Regression Testing

Client: Anonymous
Business Size: Corporation
Industry: Cash Management
Country: UK
Technology: MySQL, XML

Objective: Testing of Cash Management Systems

The Brief

The client was preparing to release new versions of their MS Windows-based cash management software that connects to a variety of cash handling machines, and their management web portal. To support this, testing of the updated systems was required to identify any defects and provide confidence that the systems behaved as required.

Methodology

The different software releases planned were both major and minor.

For the minor release, targeted testing was considered necessary for those functions modified and the associated processes. The client advised which functions needed attention. These were then thoroughly tested to confirm their correct behaviour, or identify any defects.

For the major release, a full regression test was needed. The client had an existing suite of regression test documentation that exercised each function of the system in detail, including tests covering happy flow and a wide variety of exceptions and error conditions. However, there were sections of the documentation that needed to be updated to correspond to the latest functionality. The documentation was updated appropriately, with out-of-date tests replaced according to guidance received from the client.

Some tests required MySQL queries to confirm the correct information was stored or manual examination of XML files.

Throughout all the testing, all types of defect or undesirable behaviour, from functional failures through to minor cosmetic issues, were to be recorded.

For each of the applications being tested a spreadsheet was created and shared with the client. This allowed for the recording of all defects found, identification of which version of which system was being tested, and by whom. Each function to be tested then had its own record to show:

- The function under test
- The cash machine configuration
- The application user role
- Whether the test has been performed
- Functional success or failure
- Any non-functional issue found, typically UI cosmetics
- A full description of any issues found, including severity and importance ratings

There was also space for client staff, typically developers, to record comments, ask for clarification, or declare a fix as ready for testing. Following the addition of a comment from the client, additional information would be provided or a re-test would be performed, updating the recorded results.

Challenges

There were multiple challenges to this testing work. Our consultant had not worked with cash machines before, so this was an unusual assignment. The machines work with coins and notes so the majority of the tests involved physically operating the machines, depositing or dispensing currency and provoking error conditions. There are also many different machines supported, with differing facilities, therefore understanding the available functions and the physical configuration for each machine was necessary.

Simulating error conditions could be particularly tricky, for example simulating a note jam in a well-behaved machine involved some imaginative thinking to upset operation in a way that didn't risk any damage to the expensive machinery. More straightforward was simulating power cuts during machine operation.

Even when everything is operating smoothly, confirming everything was in the correct places inside the machines wasn't trivial and often included emptying various compartments or opening them up to inspect them, and was quite time consuming.

With a range of issues found, it was up to the client to decide whether each should be addressed and fixed prior to release to their customers.

Results
As a result of the testing, and subsequent fixes applied, the opportunity for errors in physical operation or recording of transactions was significantly reduced, and numerous cosmetic improvements were made.

Lessons Learned
When testing software that controls physical hardware, there are a number of challenges that don't exist when testing a purely software systems. If interaction with those machines is necessary, remote work is impossible. Ensuring the hardware is in the expected state at the start and end of a test can be quite involved.

Also, effective communication with the client was essential, especially in the areas where the existing documentation needed to be updated.