Testing of the new integrated sample and order management software at PSI HOTLAB

MARCO STREIT¹ (AHL), COLIN KOST (IQS), MICHAEL KIEL (IQS), DANIEL SALOMON (IQS)
Paul Scherrer Institut, CH – 5232 Villigen PSI
IQS Ltd, CH – 4800 Zofingen

ABSTRACT

For the detection and control of nuclear fuel samples and monitoring with respect to criticality safety, the 1998 from PSI developed KBuch program has been used. In autumn 2011, a logic error was detected in the software during a routine booking with extraordinary nuclear fuel. This bug was reported to the national regulator ENSI and appropriate administrative measures have been taken to prevent the recurrence of the logic error.

ENSI additional requirement: “Given the fact that the logic fault is inherent and can’t be resolved in the short term, effective measures to prevent future accounting errors must be taken to medium term. The software should be developed to state of the art as soon as possible”.

Subsequently, a project was launched to update the used software as required.

The project was set up to integrate the new sample and order management (IPV) software into the Quality Management Software IQSoft used at PSI HOTLAB. Thus IQS Ltd. was chosen as project partner for the realization of the software.

After setting up an extensive catalog of requirements for the new software, PSI HOTLAB started to develop the testing procedure to ensure the needed functionalities of IPV.

During the setup of the test scripts it was found that important additional information was produced for the realization partner. A strong interaction with IQS Ltd. was created to ensure information exchange at the right time.

The used working tools and methods, as well as the status of the final software IPV is discussed.

¹ Phone +41 56 310 22 83, fax +41 56 310 22 03, marco.streit@psi.ch
1 Introduction

The department HOTLAB (AHL) of Paul Scherrer Institute (PSI) accounts, according to internal as well as external regulations, nuclear materials and moderators. The capacity of the HOTLAB is limited by various dispositions by the Swiss regulator ENSI (Eidgenössisches Nuklearsicherheitsinspektorat) and the Swiss Federal Office of Energy (SFOE).

For the detection and control of nuclear fuel samples and monitoring with respect to criticality safety, the 1998 from PSI developed KBuch program (Figure 1) has been used. In autumn 2011, a logic error was detected in the software during a routine booking with extraordinary nuclear fuel. This bug was reported to the national regulator ENSI [1] and appropriate administrative measures have been taken to prevent the recurrence of the logic error.

![Screenshot of KBuch software](image)

ENSI additional requirement: “Given the fact that the logic fault is inherent and can’t be resolved in the short term, effective measures to prevent future accounting errors must be taken to medium term. The software should be developed to state of the art as soon as possible”.

Subsequently, a project was launched to update the used KBuch software as required. Furthermore, it has been shown during the situation analysis, that there are numerous overlaps and interfaces between the used applications KBuch, another software for managing all samples and partly orders (HPI) and the Quality Management Software IQSoft [2].

The project was changed to integrate [3] the new sample and order management (IPV) software into the used the Quality Management System (QMS). Thus IQS Ltd. was chosen as project partner for the realization of the software.

After setting up an extensive catalog of requirements for the new software, PSI HOTLAB started to develop the testing procedure to ensure the needed functionalities of IPV.

2 Integrated sample and order management (IPV)

Based on the situation analysis, it was decided that the new developed software shall be an integrated part of the Quality Management System (QMS) on one hand, but on the other it has to be a standalone program, as request from the regulator.

The integration into the QMS will be carried out to:

- simplify the usage (same GUI),
- simplify the maintenance of master data,
- avoid errors due to maintenance of data at different locations (e.g. personal data).

The independency as standalone program on the other hand is essential to ensure that:

- data are not manipulated due to accidental actions of users,
- any changes in the basic functions of the program are logged,
- the new software can be operated independently of other software.

In order to ensure both, integration as well as independence, the concept is based on establish different modules.
The new “innovative KBuch” (IKB) will be a module of an integrated sample management system (IPV), which also includes a module for order and sample management (APV) and will use a part of the existing QMS, but IKB will also work as standalone software if it is necessary.

Figure 2 shows the conceptual structure of IPV showing IKB, APV and the QMS modules in the IT environment of PSI HOTLAB. All program modules will run on a virtual server in the server park of the PSI to ensure high IT security.

IPV will consist of a special databases for IKB and all other data, an encapsulated function module IKBMod (kernel) implemented to secure safety-related calculations and sample manipulation, expanded functional modules for user-friendly design of the application IKB, function modules for APV, graphical user interface for data maintenance, the database management system IQS as well as various interfaces to migrate or export data. These modules have been described in more detail in earlier publication [3].

3 Testing of Software

After setting up the specification of the new software the question appeared how AHL should test the new software. “Testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results” [4].

Of course AHL has to prove that the specifications are fulfilled, but furthermore it is necessary to check the algorithm of IKBMod in detail to ensure that the calculations are right and the used limits are correctly interpreted by IPV. So “Testing is the process of demonstrating that errors are not present” [5], but these thinking might be one causes of poor program testing.

Myers [5] proposes to add value through testing and with this raising the quality or reliability of the software, which in this case means finding and removing errors. Therefore the starting point for testing should be the assumption that the software contains errors and by testing, as many as possible of those errors should be found. Or to put it in one sentence: “Testing is the process of executing a program with the intent of finding errors.” [5]
In order to systematically plan and implement the software acceptance test (SAT) a study [6] was launched to get a good theoretical background of software testing in AHL. The process of planning the SAT begins by creating the requirements specification and ends with the actual implementation.

A test concept has been set up which reflects the general approach:

- which general test strategy is used,
- what are the aims of the test,
- which risks should be covered,
- what are the testing activities,
- which resources are available and
- which metrics are used during testing.

### 3.1 Test strategy

The test strategy defines the general approach for the testing. It describes the used testing procedures, manages the resources available and defines the quality characteristics detected and the risks to be covered. In addition, the priority of the testing activities is set within the test strategy.

The overall strategy for the IPV testing is performing all needed manipulations of the software at least once with a larger group of testers which will be the later main users of the software. But to find and eliminate most errors as early as possible the planned tests are communicated as early as possible to the IPV developers and the tests will start already with alpha- and beta versions of the software.

### 3.2 Aims of the test

The aim of the final SAT is to ensure that

- the specifications are fulfilled by the developers,
- the calculations performed by the software are correct,
- the limitation set in the program are used in a correct way by the software and
- the software finally could be released by the regulator.

### 3.3 Risks

The risk assessment of the software is an important part of the test concept as it enables the prioritization of the tests and the design of the best test sequence. The risk is calculated as the probability \( W \) of an event multiplied by the event damage potential \( P \):

\[
R = W \times P 
\]

For the probability a value between 0 and 1 is assigned. The value depends on how many error conditions or quality problems are expected and secondly on the execution frequency of the dedicated function. The number of error conditions or quality problems is not known and has to be estimated. The execution frequency refers to an estimated number a function is executed. When a function is executed once a year there is less risk to manifest a fault condition as if as a function is executed hundreds of times a day.

For the damage potential a relative value between 1 (low damage potential) and 10 (high damage potential) might be used.

A good overview of the test focus is given by entering the calculated risks in a diagram (Figure 3). Contrary to the first consideration (Figure 3 left side) to prioritize SAT dedicated to the very high risks, the test focus should be set to risks with a high potential for damage and low probability of occurrence (Figure 3 right side). Errors with very high risks should already be found during the testing of the software developers during the programming of the software, as their probability to occur is high.

One result of the study [6] was a set of special tests dedicated to the high test focus found by the risk assessment done for IPV.
3.4 Testing activities, Resources and Metrics

Testing activities are all actions dedicated to the SAT, describing what types of tests are used and which test focus is set and to what extent the SAT should be performed.

The resources need for the SAT include the existing or needed funds including human resources, finance and infrastructure.

Finally metrics are needed to make the test results measurable.

4 Testing procedure for IPV

For setting up the SAT for IPV all single requirements of the requirement specification have been implemented in a special database (Figure 4). For the SAT it is planned to begin with a clean system and by setting up the system with test data as well as real data all requirements should be tested at least once (chapter 3.2).

A series of test scripts is set up carefully and each is linked to the requirement (Figure 5), which is tested by the single script. This database of requirements and test scripts is shared with the software developer to enable IQS Ltd. to perform all tests already during their in-house testing procedure.

Due to the fact that the scripts are set up with real values to ensure the functionality of the used algorithms the software developer has the possibility to use better values for their tests than any dummy values they would normally use. A special focus is given to the tests found by the risk assessment (chapter 3.3) for IPV.
The scripts are put in a special order to optimize the testing procedure. For the SAT the scripts of the database will be printed out in a special one-page form, which then will be filled out by the dedicated tester during the SAT. The results of the test will be introduced into the database and the original signed test forms will be used to document the SAT to the regulator.

5 Lessons Learned

More than 1000 single requirement specifications have been introduced into the database as well as already more than 1000 single test scripts.

The formulation of the test scripts leads to uncovering poorly formulated requirements. Due the exchange of the test scripts with IQS Ltd. some misinterpretation of single requirements by the programmers were detected in a very early phase of the development.

The discussion of the test script with the software developer leads to a better understanding of the needs on both sides.

As lessons learned from a strong interaction between PSI HOTLAB and IQS Ltd during the project was introduced.

6 Conclusions and Outlook

The project IPV was set up to integrate the new software into the existing QMS IQSoft used at PSI HOTLAB.

After setting up an extensive catalog of requirements for the new software, PSI HOTLAB started to develop the testing procedure to ensure the needed functionalities of IPV. A large number of test scripts have been formulated and during this setup it was found that important additional information was produced for the software developer. A strong interaction with IQS Ltd. was created to ensure information exchange at the right time.

At the moment large parts of the basic setup (Figure 6) of IPV have been realized and the first pre tests have been performed.

As next step the finalization of IKBMod, the kernel to performe the safety-related calculations and sample manipulation and the test of it to request the release of IKBMod by the regulator ENSI.
Figure 6: Isotopic table of the testing version of IPV

7 Acknowledgement
The authors wish to R. Jany who gives a good theoretical basis with his bachelor thesis [6] for this paper.

8 References