METRUS
Valve Test Bench Excellence
since 1970

White Paper

Valve Test Benches as Part of Digital Process Chains

Valve Test Benches as Part of Digital Process Chains

Whether the topic is digitalization or the Internet of Things (IoT), collecting and connecting data is one of the defining issues of our time. Valve testing is involved in this in two ways. First, the testing process itself requires specific information about the valve. Second, the results generated by the test should be added to the valve's history or production data record. In a connected environment, valve testing accesses existing data and returns the generated test results to the next step in the downstream process.

Expectations and Reality

Test bench software is often used primarily as a digital display. This is certainly useful, since it provides both scales and easy-to-read numerical values. It is also much easier to switch between sensors than between conventional pressure gauges or flow sensors. However, displaying values represents only a small part of the functionality that test bench software typically offers.
As early as 2004, test bench operators were asking for “software without charts” — in other words: just put a nice, large gauge on the screen. At the same time, management and production teams were — and still are — developing ambitious software ideas and networking concepts.

The reason for the limited use of test bench software lies in the nature of the testing process itself. Without software, the tester receives a job sheet and records the test results there. Modern valve manufacturers may replace this job sheet with an input screen in the company’s software system. In either case, the tester is working with a third-party system that requires only minimal extra effort beyond the actual test procedure. Everything else is outside the operator’s immediate focus.

When test bench software is not integrated into a connected environment, a significant administrative burden is shifted to the test bench. The tester is now expected to enter valve data correctly and produce a clean, signed test report at the end. The task becomes much more complex, because the tester has effectively become a software operator as well. The additional time required creates costs that are not normally included in the testing budget. On top of that, the tester now carries additional responsibility for documents and their accuracy. As long as there is still an authoritative third-party system running in parallel to the test bench software — such as a job sheet or ERP system — it is only natural that the test bench software will be used merely as a digital display.

A New Approach

Until now, test bench software has mainly been designed around the scope of the measurement task — the more functions, the better. In practice, however, many of these features play little role in day-to-day testing and are typically used only by development departments and quality assurance teams. To deliver maximum value, the software solution must meet not only the needs of the test itself, but also the requirements of different stakeholder groups within the company.

For test bench operators — less is more: as simple as a smartphone app

1. Minimal data entry
2. Minimal number of clicks
3. Minimal number of decisions

For cost accounting — minimal costs
4. Minimal time delays
5. Use of existing data
6. Minimal rework

For quality assurance — clear, reliable processes with little room for operator discretion
7. Defined rules for how results are determined
8. Defined pass/fail criteria

For document management
9. Completed reports ready for end-customer documentation

Today, networks and database systems are standard technology. Information about the test object is already available. Even in service environments, Excel spreadsheets with test object data are often available. Integrating the test bench into the digital process chain through its software, using existing data directly, and returning results and reports to the process chain is the logical next step in the evolution of test bench software.

To make this work — and to meet the needs of all stakeholder groups — three challenges must be addressed:

I. Connection to existing data sources
II. Return of completed digital reports and relevant data
III. Adaptation and partial automation according to customer needs

These challenges can be solved through two standard technologies and an appropriate software architecture. The basis is METRUS CRS test bench software, which comprehensively handles the measurement task, includes a database, and generates test reports. The software is built entirely as a scripting solution and can therefore be adapted extensively to individual requirements. For communication with external data sources, web services are an integral part of the software architecture.

Connection via Web Services

Almost all test bench software solutions on the market include a database. In connected data environments, it has therefore been technically possible for many years to exchange data between test bench software and company databases at database level. The reason this usually does not happen lies in the complexity of inter-database communication. The required technologies belong to the highest level of programming and system administration. For the test bench environment, this barrier is generally too high.

This hurdle can be bypassed by implementing the widely used internet technologies web services and XML as an additional data access option. METRUS CRS can access both its own internal database directly and external company databases via web services.

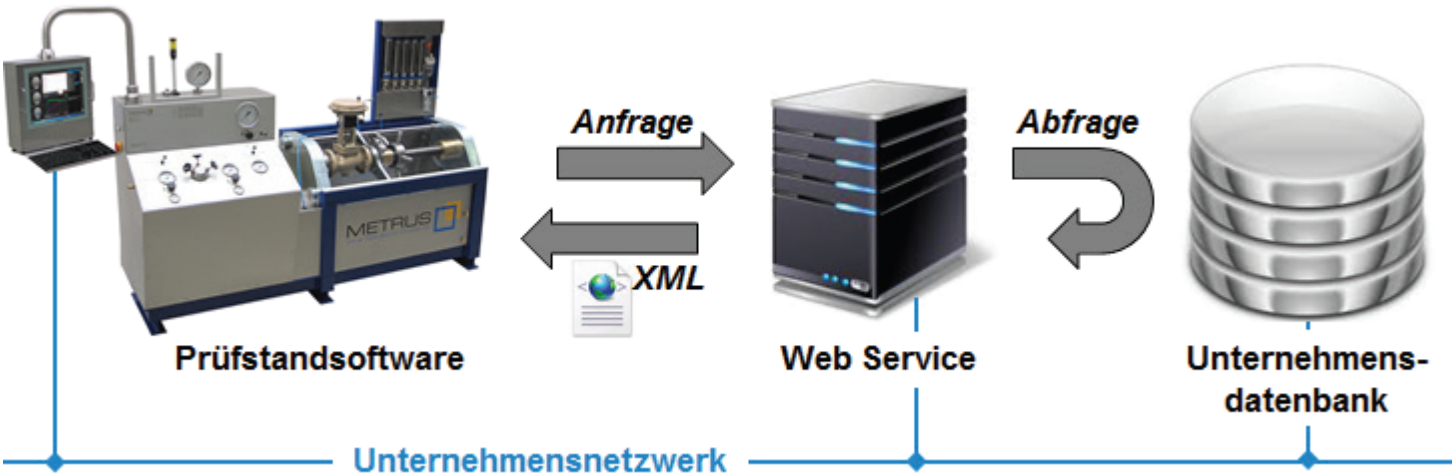

Figure 1: Access to corporate data via web services

. The process works as follows:

1. The tester enters, for example, the order number or valve ID at the test bench and clicks the “XML Import” button.
2. The METRUS CRS software sends this search data to the web service.
3. The web service retrieves all relevant details from the company database and returns them to the METRUS CRS software in the form of an XML file.

The imported data typically includes valve characteristics, limit values, and additional order data for the test report. The principle is similar to the issue desk in a library. The reader (METRUS CRS) decides which book is needed and tells the librarian at the desk (the web service). The librarian knows the endless shelves of the library, finds the book, and hands it over.

Just as the reader at the issue desk does not know the library’s internal filing system, METRUS CRS does not need to know the structure of the company database from which the information is retrieved. No matter which system the data comes from, the workflow in METRUS CRS always remains the same. In other words, no custom programming is required to connect the test bench software. It is sufficient to enter the web service connection details in the designated location. As lean internet technologies, web services and XML are ideally suited for use in widely distributed corporate networks with multiple locations.

The Web Service as an Administrator’s Progect

The web service itself consists of two parts: a web server and a PHP script running on it. A web server is a standard software component that can run on one of the company’s servers. The widely used Windows Server operating system already includes IIS (Internet Information Services) as a web server at no additional cost. In other words, the web server is usually already available or can be deployed free of charge on other server operating systems as well.

The PHP script is a text file containing a list of commands and instructions. These are executed by the web server. Just as an MS Word document requires Microsoft Word in order to be edited, a PHP script requires a web server in order to run. “PHP” is simply the name of the scripting language, comparable to Java or Visual Basic.

The only customer-specific task is to create the PHP script that accesses the company database and generates an XML document from the retrieved data. Since the exact XML format is predefined by METRUS CRS, the task is essentially reduced to finding the desired information in the company database.

PHP scripting is part of the skill set of many administrators and basic knowledge for any web developer. The expertise is therefore often already available within the company or can be sourced from web developers at manageable cost. As a result, the project of connecting the test bench to the company database becomes a manageable in-house administrative project.

.

Minimalism as a Requirement

Once the test bench software has been successfully connected to the company database, the operator no longer needs to enter data manually. This primarily saves time and effort and therefore reduces costs. However, the complexity of the task is only partly reduced by this alone. Ease of use is just as important for software acceptance. The goal is to minimize the required number of clicks and, above all, the number of decisions the operator has to make. The software should include only those functions and decisions that are actually necessary for the specific application — less is more.

Traditional test bench software usually offers the maximum possible range of functions, from which users select the features they find useful. Special requirements often result in custom software projects, which typically exceed the budget allocated for the test bench itself. In METRUS CRS, the need for highly specific functions on the one hand and a software solution reduced to the essentials on the other is addressed through a special architecture. The user interface and all calculations are implemented through scripts and are therefore not fixed in an unchangeable executable file. A script is simply a file that can be opened and edited with any standard text editor. This means that adapting calculations or functions does not require separate custom software, but only a slightly modified script.

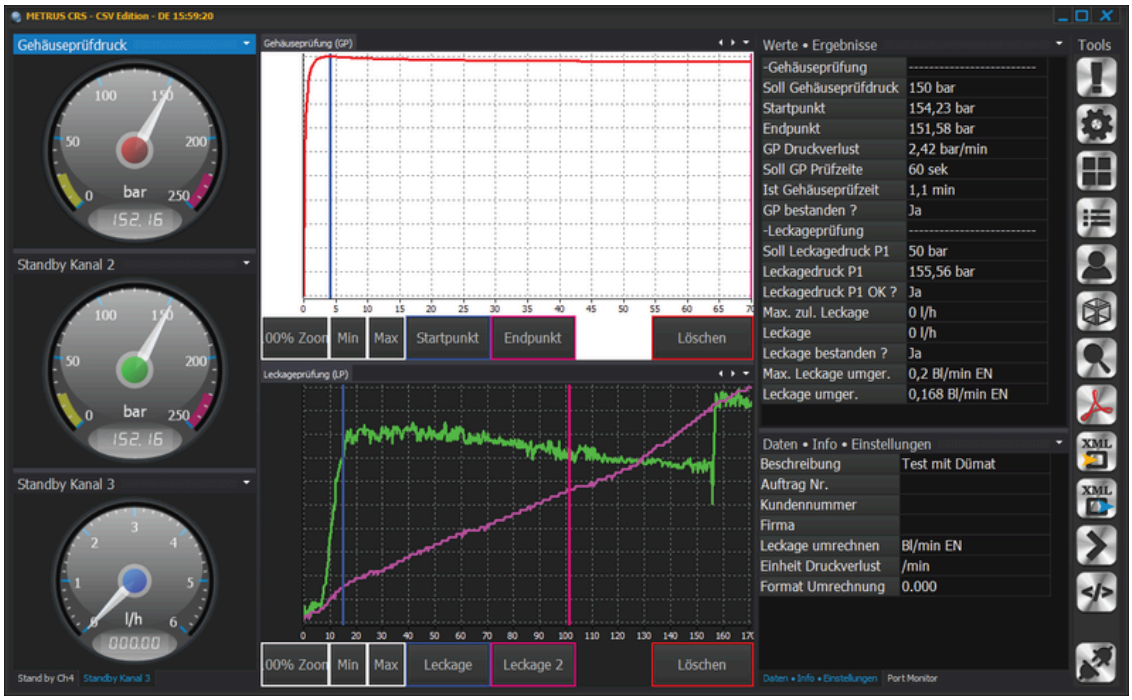

Figure 2 shows the standard scope of METRUS CRS as a classic standalone solution.

.

The tester manually enters all data and test parameters, starts the recording, evaluates the chart manually after the test, saves the record, and finally generates a PDF file, which is then stored under a specific file name in a defined location. This process involves moving through several input and selection screens.

In the first step, all relevant details and limit values are loaded from the company database via web services, and the test parameters are calculated automatically. The script before recording starts determines the recording duration.

The script after recording ends sets the selection points. In the case of a body test, for example, this means setting the start point to the maximum value and the end point to the last value in the recording. Figure 3 shows the optimized approach based on data integration and scripting. The tester enters the valve serial number and clicks the XML button. The recording is started, the test is carried out at the bench, and finally the tester clicks the PDF button — done. Throughout the entire process, only one software screen is used, with a single data entry and four buttons.

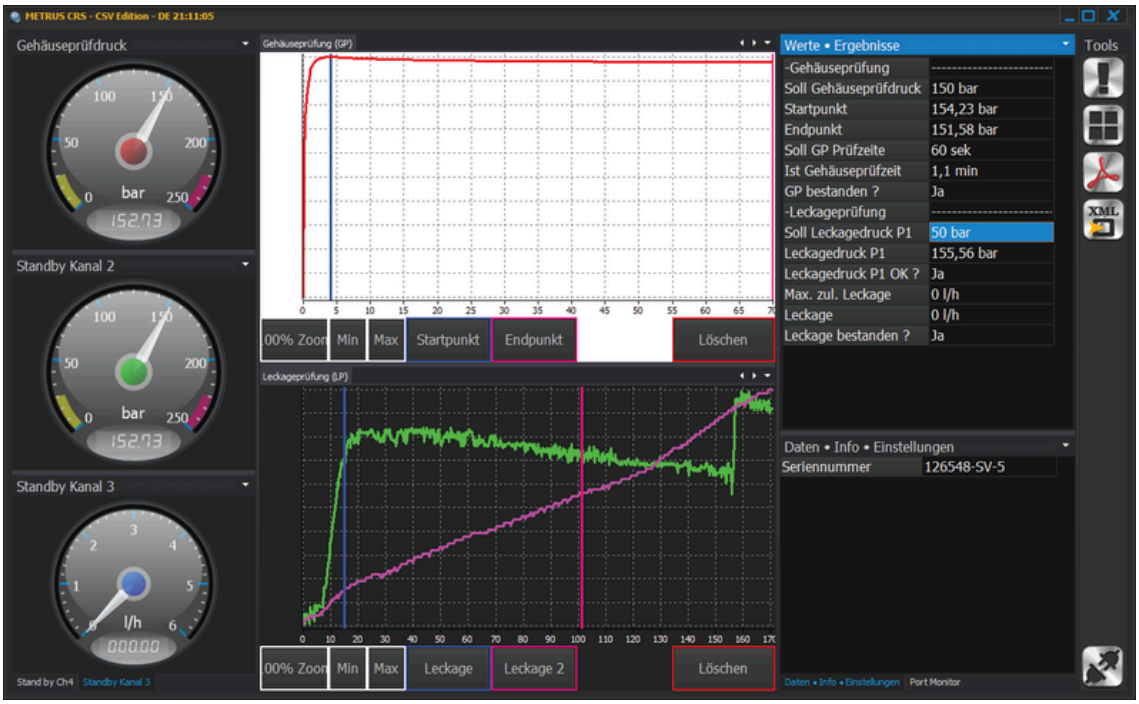

Figure 3: METRUS CRS – with data connectivity and reduced to essential functions.

.

Clicking the PDF button also triggers a script that executes the following steps in the background:

1. Check whether all tests have been passed; if not, abort with a message
2. Generate the PDF test report
3. Create the file name from the content of selected data fields
4. Save the PDF file in a predefined network path
5. Save the test data record as a log in the local METRUS CRS database
6. Display the message: “Report saved”

The scripts used here are standard components of the software. They are not created specifically for data integration; they are simply modified as needed.

The PDF Test Report as Part oft he Downstream Process Chain

The final digital result of valve testing is a report in PDF format. If designed appropriately, this report can become a direct part of the valve documentation delivered to the customer. However, the digital process chain does not always end there. Depending on the complexity of the requirements, two further steps may be necessary: digital signature and return of suitable test data.

Many companies still print test reports on paper, sign them, and then scan them back in. Because the test report has document status, it becomes valid only once it has been signed by the tester and, where required, countersigned by an inspector. To ensure that the digital process chain does not break at this point, METRUS CRS includes a digital signature function.

Figure 4: Digital signature via signature pad

.

When this function is used, a window opens immediately before the PDF file is generated. The tester signs on the signature pad and confirms the input. A second signature field is available for inspectors or customer acceptance procedures. The result is a digitally signed test report.

In comprehensive company database systems with integrated document management, the digital process chain is only complete once the successful completion of the test has been reported back to the system. As with data import, a web service is also used here for data return, configured by the user according to specific requirements. METRUS CRS sends an XML document to the web service, which writes the contained information into the company database. Particularly relevant here are the path to the PDF test report, together with the valve ID, order number, and test date. These data provide direct access to the test report from within the company software.

METRUS GmbH
Richard-Lucas-Str.6
41812 Erkelenz
Germany
E-Mail
sales@metrus.de
Contact
+49 (2431) 94502-0

Data Import from Excel – the „Small-Scale“ Solution

A web service is highly efficient when valve and order data are already available in a company database system. Especially in service environments, however, this is often not the case. Even so, the benefits of scripting technology and automation — from the start of the test to the finished report saved on the server — offer the same advantages here as they do for large companies.

In this case, the data import software FlowHeater represents the first step in the digital process chain as a manual process. The work preparation team formats the available valve data and, if necessary, the customer list into a suitable Excel format and then uploads it via the network into the METRUS CRS database. Data entry at the test bench is then reduced to selecting the valve and customer from a list. From that point onward, the workflow is the same as with the direct database connection via web services described above.

The networking of digital systems secures the medium- and long-term competitiveness of every company. As the final step in valve manufacturing and maintenance, the test bench is therefore ideally suited to become an integral part of the digital process chain.