This article presents a discussion on the considerations around the use of automated test tools vs. manual testing and provides an example of a calculation of Return on Investment (ROI). This article complements and expands on the information contained in the revised GAMP Good Practice Guide on Testing of GxP Systems, which is currently under development.

### Background

In 2005, ISPE released the GAMP Good Practice Guide on Testing of GxP Systems. This Guide was provided electronically as a download available free of charge to ISPE members. It was made available in this format in recognition of the provisional status of the document, which captured the initial thoughts and concepts around testing GxP systems.

During the years since the document was written, there has been an increased focus on the use of risk-based approaches in industry, accompanied by a significant regulatory and industry activity affecting computerized systems, and much evolution in testing good practice, including:

- GAMP 5
- ICH Q8, Q9, Q10
- ASTM E2500
- Adoption and implementation of Process Analytical Technology (PAT) and Quality by Design (QbD)
- Increased industry focus on risk-based approaches
- Increased use of non-linear development lifecycles
- Increased use of computerized test tools
- Revised EU GMP Annex 11

A second edition of the Guide has been in preparation for the last year with the document substantially revised to provide definitive guidance around effective, risk-based testing of GxP systems. The new Guide contains pragmatic guidance on defining the scope of testing, the strategy, the test plan, recording the results, dealing with deviations and reporting the outcome. The central premise of the Guide is that the aim of testing is to discover defects in order to mitigate risk. It covers conventional linear sequential development methodologies as well as the iterative and incremental approaches. Practical guidance is included on the selection, benchmarking, assessment, control and use of automated testing tools. Appendices provide worked examples for different types of systems, focusing on the risks associated with any unique features of the system type along with suggestions on how to mitigate the risks. The examples cover systems applying PAT, cloud computing, packaged systems, analytical instruments, infrastructure, process control systems, configurable IT systems, and end user developed applications.

The GAMP D-A-CH Special Interest Group (SIG) on Test Automation, in combination with the GAMP Americas SIG on Test Automation, provided the key content of Appendix T11 “Automated Test Execution and Computerized Test Management Tools” in the upcoming second edition of the GAMP Good Practice Guide (GPG) on Testing of GxP Systems, which is currently in final review.

In this article, the members of the GAMP D-A-CH SIG on Test Automation present a discussion on the considerations around the use of automated test tools vs. manual testing and provide an example of a calculation of Return on Investment (ROI). This article complements and expands on the information contained in the Good Practice Guide.

### Introduction

The SIG is neither recommending nor advocating for test automation in general, but aims to highlight the benefits and risks of test automation, point out the differences between manual and automated testing, define criteria for selection and guidance for validation of test automation tools, and provide a model for calculating the ROI. While most aspects are fully covered in Appendix T11 of the above-mentioned GPG, the criteria and methods to calculate the ROI are unique to this article.
A key concept to consider before we start is the difference between two types of testing tools:

- **Test Management Tools** facilitate the task of test case and test script authoring, requirements management, review and approval, test execution, and test deviation management.
- **Test Automation Tools** assist in authoring or recording test scripts, automatically executing test scripts, and recording test evidence.

While this article focuses on tools for automated test execution, decision criteria for test management tools are addressed as well to complete the picture (see sidebar “Decision Criteria for Test”). Performance test tools that are being included in the GAMP Good Practice Guide on Testing of GxP Systems are not considered here.

### Calculating the ROI for Test Automation Tools

An important aspect to be considered when planning the use of test automation tools is a calculation or estimation of the cost effects. These typically include initial costs for the assessment and set-up of the tool, the implementation of the tool infrastructure, training, and the initial creation of automated test scripts. In addition, specific parameters should be considered when calculating the return on investment.

This article gives an overview of the calculation and a description of a basic ROI model with further guidance on cost effects specific to regulated industries such as life sciences. In general, any suitable ROI model should assist in estimating the expected gains (or losses) for using test automation tools, especially when comparing costs and benefits to traditional manual testing. Regulatory requirements as well as industry-specific risks typically require a greater amount of testing and documentation of the software used to verify computerized systems, and this should be taken into account.

ROI calculation models can be found in various publications as well as in books and on the internet. These models typically take into account the more obvious factors, e.g., comparing the effort (or costs) for creating and executing automated test cases plus tool license costs versus the effort (or costs) for creating and executing manual test cases.

However, more complex and sometimes “hidden” costs need to be considered as well, e.g., follow-up costs caused by insufficient test coverage or failing to address multiple scenario boundary tests. These additional costs can often be reduced—or even completely avoided—by applying a comprehensive automated test suite with appropriate coverage. In addition to avoiding additional costs, a test suite also may help to increase confidence around excluding certain test cases. These additional factors have been included in the ROI model suggested by this article.

### Manual vs. Automated Testing

A key decision significantly impacting the test strategy as well as the cost calculation is what types of tests will be executed, and of those, which will be automated. Obviously, some test types are more suitable for automation than others. Some tests cannot be automated at all except in combination with emulations or mock-ups (e.g., those involving interaction with real hardware, such as balances, scanners, or other devices), whereas other tests rely on test automation and cannot be executed manually (e.g., load and stress tests).

Even more complex, a good ROI calculation needs to be able to handle mixed test modes as well, i.e., scenarios in which some tests are executed manually and some are automated. In real life scenarios, test automation often starts in a setup with existing manual tests, and consecutively automated tests either replace or complement the existing suite, so the ratio of manual vs. automated tests changes over time, impacting the ROI calculation. Test types that should be considered in different formulas include, but are not limited to those listed in Table A.

#### Formula – How to Calculate the ROI

The types of tests shown in Table A are usually related to different use cases or different life-cycle phases of the Application Under Test (AUT). This can be reflected in a powerful, yet flexible formula that needs to be adapted to a specific project and its current stage, thereby allowing to calculate the ROI per test type and under various constraints. It allows a stepwise or staged approach when introducing test automation into an existing manual testing scenario.

**General Calculation:** *Gain vs. Investment:*

The general formula for calculating the ROI is as follows:

\[
\text{ROI} = \frac{\text{Benefit}}{\text{Investment}} = \frac{\text{Gain} - \text{Costs}}{\text{Investment}}
\]

As expected, the ROI will be positive if the gain (here: savings)
Test Automation ROI

exceeds the costs (that include the investment), and it will be negative if the costs are greater than the gain. In both cases, the relative difference between gain and costs in relation to the overall investment will determine the ROI.

Outside the life sciences industries, generally understood benchmarks are that automated test tools typically pay for themselves (achieve break even) by the second or third round of test execution (typically regression testing or development testing when used in non-linear software life cycles).

The initial direct costs of setting up an automated test tool are greater in the life sciences industry, because of the additional costs to assess, document, and control the tool. This typically achieves break even by the third or fourth testing cycle for new applications with the variation being due to various parameters discussed below.

Costs to Consider

When calculating the ROI for automated testing, both direct and indirect costs need to be taken into account. Direct costs are directly related to testing, whereas indirect costs originate from errors that have not been detected (e.g., increased support, bug fixing, recalls etc.). Obviously, the indirect costs are much harder to determine and can often only be estimated.

This section discusses various parameters and some suggestions that should be considered when calculating the ROI (as shown above). Although the basic formula is simple, i.e., \( \text{ROI} = \frac{\text{Benefit}}{\text{Investment}} \), it can be quite challenging to properly calculate (or estimate) the gains and the investments.

Direct Costs

Tool costs (one-off costs relevant to the test automation tool):

- Tool acquisition and assessment costs

<table>
<thead>
<tr>
<th>Factor</th>
<th>Description</th>
<th>Application-specific</th>
<th>Life Sciences-specific</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total # of test cases</td>
<td># of tests to be executed (manual and automated)</td>
<td>Yes</td>
<td>No, but typically the test coverage is above average</td>
<td>N</td>
</tr>
<tr>
<td>Total # of automated test cases</td>
<td># of tests intended for automation</td>
<td>Yes</td>
<td>No</td>
<td>I</td>
</tr>
<tr>
<td># of executions per cycle</td>
<td>Average # of tests executed in each cycle</td>
<td>Yes</td>
<td>Yes, typically more repetitions are needed</td>
<td>G</td>
</tr>
<tr>
<td># of testing cycles</td>
<td># of cycles during ROI evaluation time (a cycle may be a week, a build, release, etc.)</td>
<td>Yes</td>
<td>No</td>
<td>G</td>
</tr>
<tr>
<td>Tester hourly rate</td>
<td>The average tester salary</td>
<td>No</td>
<td>No</td>
<td>G</td>
</tr>
<tr>
<td>Failure cost</td>
<td>The predicted cost related to failures in production application functionality</td>
<td>Yes</td>
<td>Yes, with product recalls as the worst case scenario</td>
<td>G</td>
</tr>
<tr>
<td>Manual test execution and analysis time</td>
<td>The average time it takes to execute a manual test and analyze the results</td>
<td>Yes</td>
<td>Yes, typically documentation overhead is above average</td>
<td>G</td>
</tr>
<tr>
<td>Tool acquisition, assessment, and license costs</td>
<td>Test tool costs</td>
<td>No (general)</td>
<td>No, although specific assessment activities may be needed</td>
<td>I</td>
</tr>
<tr>
<td>Tool training costs</td>
<td>Cost of training resources to use the test tool</td>
<td>No (general)</td>
<td>No, although qualification and training may require effort above average</td>
<td>I</td>
</tr>
<tr>
<td>Test machine cost</td>
<td>Cost of machines used for unattended execution of automated tests (separate from machines used for manual test execution)</td>
<td>No (general)</td>
<td>No, standard hardware may be used</td>
<td>I</td>
</tr>
<tr>
<td>Test automation environment activity period</td>
<td>The period of time that the test automation environment (incl. tools, machines, etc.) will be in use</td>
<td>No (general)</td>
<td>Yes, typically applications will be used for longer than average</td>
<td>G</td>
</tr>
<tr>
<td>Test development/debugging time</td>
<td>Script development time</td>
<td>Yes</td>
<td>Yes, typically additional documentation and care is needed</td>
<td>I</td>
</tr>
<tr>
<td>Automated test execution time</td>
<td>The average time it takes for a single automated test to execute</td>
<td>Yes</td>
<td>No</td>
<td>N</td>
</tr>
<tr>
<td>Automated test cycle analysis time</td>
<td>The average time it takes to analyze the results of one full test cycle execution</td>
<td>Yes</td>
<td>No, although additional documentation and care is needed</td>
<td>I</td>
</tr>
<tr>
<td>Automated test cycle maintenance time</td>
<td>The average time it takes to perform script maintenance following each full test cycle execution</td>
<td>Yes</td>
<td>Yes, typically additional documentation and care is needed</td>
<td>I</td>
</tr>
<tr>
<td>Overnight execution?</td>
<td>Indicates whether tests are executed during the day, or also overnight</td>
<td>No</td>
<td>No</td>
<td>N</td>
</tr>
</tbody>
</table>

Table B. Parameters for ROI calculation for automated test tools.
Test Automation ROI

- Licenses for a test automation tool
- Initial training of core test automation team
- Initial configuration of the tool

Application-specific costs (relevant to each application being tested):

- Training of project-specific staff members
- Creation of a test automation framework with general functions/classes
- Creation of test data
- Creation of automated test cases
- Maintenance of automated test cases
- Execution of automated test cases
- Creation, maintenance, and execution of manual test cases (typically not all test cases will be automated)

Indirect Costs
Indirect costs may be included in the ROI calculation, too. Often, these costs need to be estimated based on experience from similar projects in the past:

- Increased risk of not detecting a failure before the application is used in production
- Costs originating from undetected failures

Parameters – Create Your Own Formula
The following parameters have initially been taken from a generally available ROI Calculator, but have been analyzed, extended, and adapted for applications used in regulated industries. Three columns, “Application-specific,” “Life Sciences-specific,” and “Direction,” as seen in Table B were added to provide guidance on which aspects to include in the ROI calculation and which ones to emphasize.

The added columns have the following meaning:

- Application-specific: is the factor specific for one application or is it of a more general nature and should therefore be distributed over several applications?
- Life Sciences-specific: parameters that require specific attention for applications used in regulated environments. This can be reflected by a relative weight in the formula.
- Direction: specifically, when automating tests, this factor is likely to increase Gains (G) or contribute to the Investments (I). In case that the difference is negligible, the factor is rated to be Neutral (N).

In Table B, the first three rows are used to explain the concept: The total number of test cases obviously depends on the Application Under Test (AUT), but is not necessarily life sciences-specific, and the number does not influence the ROI calculation, so it is neutral. The total number of automated test cases is also influenced by the AUT, and again regulatory aspects have no significant impact. Obviously, with more tests being automated, the investment in automation increases, as every test case requires specific effort, so when calculating the ROI, this is clearly an investment. However, with a higher number of executions per cycle, this investment in test automation will pay off so it will become a gain. The factor depends on the AUT (e.g., on the frequency and amount of changes), and with higher quality standards, more repetitions can be expected.

Example – Dispense Application
With the help of the direct and indirect costs and the parameters, the simple formula shown in “Formula – How to Calculate the ROI” can now be defined in more detail to calculate the ROI of test automation for a specific application.

*The example as seen in Table C is based on having a dispense application (either stand-alone or as an integrated part of a more comprehensive Manufacturing Execution System (MES)) that has already been validated. Now a number of changes and enhancements are foreseen, so we wonder if automating a fair share of the existing test suite would be a good idea. In order to set up the formula and calculate the ROI, we need to act on a number of facts and assumptions. To simplify the example, not all the factors in Table B are used in Table C.*

Starting with an existing suite of manual tests, the initial costs of executing one testing cycle can be calculated by using the numbers from above:

Test Execution Costs, manual only, per cycle =

\[ EpC_{\text{total}} \times \text{Rate} \times \text{Time}_{\text{man}} = 25 \times \frac{\$50}{\text{hr}} \times 4 \text{ hrs} = \$5,000 \]

It also can be assumed that one final test cycle with all available tests will be executed:

<table>
<thead>
<tr>
<th>Factor</th>
<th>Abbreviation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total # of test cases</td>
<td>NoTCTotal</td>
<td>45</td>
</tr>
<tr>
<td>Total # of automated test cases</td>
<td>NoCTauto</td>
<td>36</td>
</tr>
<tr>
<td>Total # of manual test cases</td>
<td>NoCTman</td>
<td>9</td>
</tr>
<tr>
<td># of executions per cycle</td>
<td>EpCtotal</td>
<td>25</td>
</tr>
<tr>
<td># of executions per cycle, automated test cases</td>
<td>EpCauto</td>
<td>21</td>
</tr>
<tr>
<td># of executions per cycle, manual test cases</td>
<td>EpCman</td>
<td>4</td>
</tr>
<tr>
<td># of testing cycles</td>
<td>Cycles</td>
<td>variable</td>
</tr>
<tr>
<td>Tester hourly rate</td>
<td>Rate</td>
<td>$ 50/hr</td>
</tr>
<tr>
<td>Manual test execution and analysis time</td>
<td>TimeMan</td>
<td>4 hrs</td>
</tr>
<tr>
<td>Tool acquisition, assessment, and license costs</td>
<td>Coststool</td>
<td>$ 15,000</td>
</tr>
<tr>
<td>Tool training costs</td>
<td>TrainTool</td>
<td>$ 8,000</td>
</tr>
<tr>
<td>Test machine cost</td>
<td>HW</td>
<td>$ 3,000</td>
</tr>
<tr>
<td>Test development/debugging time</td>
<td>TimeDev</td>
<td>8 hrs</td>
</tr>
<tr>
<td>Automated test cycle analysis time</td>
<td>TimeAutoAnal</td>
<td>0.5 hrs</td>
</tr>
<tr>
<td>Automated test cycle maintenance time</td>
<td>TimeAutoMaint</td>
<td>0.5 hrs</td>
</tr>
</tbody>
</table>

Table C. Example calculation.
Test Automation ROI

Test Execution Costs, manual only, final test run =

\[ \text{NoTC}_{\text{total}} \times \text{Rate} \times \text{Time}_{\text{man}} = 45 \times \frac{\$50}{\text{hr}} \times 4 \text{ hrs} = \$9,000 \]

Both formulas can be combined so the costs of purely manual test execution can be calculated, depending on the number of testing cycles:

Test execution costs with all tests being manual =

\[ \$9,000 + (\text{Cycles} - 1) \times \$5,000 \]

The values returned for test execution costs with all tests being manual are in Table D as Costs \(_{\text{Manual}}\).

For the calculation of the costs including test automation, again take the numbers from Table C, assuming that 36 of the overall 45 test cases will be automated. First, calculate the effort of the remaining (nine) manual test cases by using the above formulas. Assume that 21 of the 25 tests executed on average per cycle are made automated, leaving four manual tests:

Test Execution Costs, manual tests, per cycle =

\[ \text{EpC}_{\text{man}} \times \text{Rate} \times \text{Time}_{\text{man}} = 4 \times \frac{\$50}{\text{hr}} \times 4 \text{ hrs} = \$800 \]

For the final validation cycle, there are nine manual test cases (45 – 36) left:

Test Execution Costs, manual tests, final test =

\[ \text{NoTC}_{\text{man}} \times \text{Rate} \times \text{Time}_{\text{man}} = 9 \times \frac{\$50}{\text{hr}} \times 4 \text{ hrs} = \$1,800 \]

In addition, the costs for test automation need to be included. Start with the initial investments...

Initial Test Automation Investment = \( \text{Costs}_{\text{AutoInit}} \times \text{Train}_{\text{AutoInit}} + \text{HW} \) = \$15,000 + \$8,000 + \$3,000 = \$26,000

...and the costs for the initial implementation:

Initial Test Automation Implementation = \[ \text{NoTC}_{\text{Auto}} \times \text{Rate} \times \text{Time}_{\text{dev}} = 36 \times \frac{\$50}{\text{hr}} \times 8 \text{ hrs} = \$14,400 \]

The sum of Initial Test Automation Investment and Initial Test Automation Implementation are listed at Costs \(_{\text{Invest}}\) in Table D.

For the calculation of the test automation costs per cycle, use the 21 automated tests on average:

\[ \text{Time}_{\text{auto}} = \text{Time}_{\text{AutoFollowUp}} + \text{Time}_{\text{AutoMaint}} = 0.5 \text{ hrs} + 0.5 \text{ hrs} = 1 \text{ hr} \]

Test Automation Costs, per cycle =

\[ \text{EpC}_{\text{auto}} \times \text{Rate} \times \text{Time}_{\text{auto}} = 21 \times \frac{\$50}{\text{hr}} \times 1 \text{ hr} = \$1,050 \]

Finally, the numbers of manual and automated test execution need to be added to calculate the total costs per testing cycle:

Total Test Costs, per cycle = \$800 + \$1,050 = \$1,850

Again, both numbers can be combined easily:

Automated Test Execution Costs (21 auto, 4 manual per cycle) = \$1,800 + (\text{Cycles} - 1) \times \$1,850

Table D. Example costs for different scenarios.

<table>
<thead>
<tr>
<th>Cycles</th>
<th>Costs(<em>{\text{Auto}}) + Costs(</em>{\text{Invest}}) = Costs(_{\text{Total}})</th>
<th>Costs(_{\text{Manual}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>$ 9,200 + $ 40,400 - $ 49,600 &gt; &gt; $ 29,000</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>$ 18,450 + $ 40,400 - $ 58,850 &gt; $ 54,000</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>$ 22,150 + $ 40,400 - $ 62,550 &lt; &lt; $ 64,000</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>$ 27,700 + $ 40,400 - $ 68,100 &lt; &lt; $ 79,000</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>$ 36,950 + $ 40,400 - $ 77,350 &lt; &lt; $ 104,000</td>
<td></td>
</tr>
</tbody>
</table>

Example calculation 1:

\[ \text{ROI}_{10} = \frac{\text{Gain}}{\text{Investment}} = \frac{\$54,000 - \$58,850}{\$40,400} = \frac{-\$4,850}{\$40,400} = -12\% \]

Example calculation 2:

\[ \text{ROI}_{20} = \frac{\text{Gain}}{\text{Investment}} = \frac{\$104,000 - \$77,350}{\$40,400} = \frac{\$26,650}{\$40,400} = +66\% \]

In this example calculation, the breakeven will be at 12 cycles. This seems to be in contrast to Figure 1, but this number will go down quickly for all following applications, as the initial investment for licenses, training, and hardware is a one-time effect (see formula Initial Test Automation Investment).

Also, a conversion of existing tests has been assumed, but the ratio will become much more favorable for all new test cases, as here the cost for creating manual tests is contributing to the equation. Furthermore, costs that could back up the investment such as missed failures that made it to production are not included either. If only one issue were to be detected by automated tests that would otherwise have been missed, test automation becomes attractive very rapidly.
Test Automation ROI

Figure 1. Typical Return on Investment (ROI) from the use of test automation tools.

Conclusion
In a world that is becoming more and more flexible and agile, automated test execution should be considered for applications used in regulated environments. While the initial effort seems – and often is – rather high, an automated test suite can provide numerous advantages as detailed in Appendix T11 of the upcoming second edition of the Good Practice Guide “Testing of GxP Systems.” Although neither all benefits nor all negative aspects are quantifiable, it is worthwhile to calculate if – or when – investing in test automation will pay off. This article describes a general formula that can be adapted to any specific application or scenario, using direct and indirect costs and several parameters. With some experience, project or test managers will be able to estimate the investment needed and predict the point at which test automation will become profitable. The crucial aspect, of course, is to pay attention to all factors that contribute to the overall equation.

A spreadsheet containing the ROI model discussed in this website will be posted on the GAMP COP website for ISPE members to download and utilize.

Acknowledgements
The authors would like to thank Bernhard Kausler (ITQ), Thierry Dietrich (Arcondis), and Andreas Hengstberger (Nycomed), members of the GAMP D-A-CH SIG on Test Automation, and Radha Ramesh, lead of the GAMP Americas SIG on Test Automation, for contributing to and reviewing this article. Special thanks to David Stokes (Business and Decision Life Sciences) for providing Figure 1 and helping with the review and to Charlie Wakeham (Pall Life Sciences, member of GAMP European Steering Committee) for writing the background and helping with the review.

References


About the Authors
Stefan Münch is currently Campus Quality Manager, leading the quality and test team of Rockwell Software Karlsruhe. He has more than 15 years of working experience in leadership roles in life sciences for MES applications. As an active member of ISPE GAMP D-A-CH, he was a member of the SIG Open Source Software and co-author of an article in PE magazine, and is currently leading the GAMP DACH SIG on Test Automation. Münch was a speaker at ISPE Conference 2006 and held a workshop on test automation at a German GAMP conference 2010. He graduated in computer science at the University of Karlsruhe, Germany. He can be contacted by email: smuench@ra.rockwell.com.


Peter Brandstetter is Senior Manager for Computer System Validation and Life Sciences Information Management at Arcondis GmbH in Germany since May 2011. He is also responsible for the Southern Germany and Austrian territory. Brandstetter has more than 15 years of experience as consultant and project management in computer validation, quality management/quality assurance, research and development and supply chain management in the life sciences industry. He is leading computer validation seminars and is lector for electronic document management in the pharmaceutical industry at the University of Applied Sciences, FH Campus Wien. Brandstetter is co-author of the ISPE Good Practice Guide “Risk-Based Approach to GxP Compliant Laboratory Computerized Systems (Revision).” He can be contacted by email: peter.brandstetter@arcondis.com.

Arcondis GmbH, Mergenthalerallee 79-81, 65760 Eschborn, Germany.

Konstantin Clevermann is a Senior Expert in pharmaceuticals and life Sciences at Software AG – IDS Scheer Consulting GmbH since 2002. In this role, he is responsible for compliant business processes, IT risk and compliance, IT infrastructure qualification, qualified IT operations, process-oriented SAP validation, and test management. As an active member of the International Association for...
Pharmaceutical Technology (APV), Clevermann is currently leading the APV focus group Information Technology. He is also a member of the GAMP D-A-CH SIG on Test Automation and a frequent speaker at conferences and workshops. Clevermann holds a diploma degree mathematician of Ruhr-Universität Bochum. He can be contacted by email: konstantin.clevermann@ids-scheer.com.

IDS Scheer Consulting GmbH, Niederkasseler Lohweg 189, 40547 Duesseldorf, Germany

Oliver Kieckhoefel is a Senior Consultant for regulatory affairs and advises pharmaceutical customers on how to validate or upgrade their computer system, especially SAP ERP or Documentum. Kieckhoefel has more than 20 years of experience in or with pharmaceutical companies globally. He has been validation officer for various SAP R/3 upgrade projects as well as site computer validation lead during two worldwide SAP rollouts. As a site computer validation coordinator, he was responsible for regulatory inspections by local authorities or the FDA in Germany. Prior to his current role, he worked as a software engineer at a German MES supplier in Lueneburg. His educational background is microelectronics/communication technology as a Certified Engineer (Diplom-Ingenieur). He holds a degree in Nachrichtentechnik from Fachhochschule Kiel. He frequently acts as a speaker or moderator for computer validation courses for ISPE or VDI/VDE in Germany. He can be contacted by email: oliver.kieckhoefel@btconsult.de.

btconsult GmbH, Europaallee 27-29, D-50226 Frechen, Germany

Ernst Reiner Schäfer is currently Managing Director of Abacon Gesellschaft für Mess-, Steuer- und Regelungstechnik mbH, Switzerland. From 2008 until 2012, he was in charge of project management and validation issues at Abacon. Before that, he had been with AstraZeneca for nine years, where he held the positions of system administrator, head of electrical engineering, and head of project department. From 1988 until 1999 he worked for Mannesmann, Hartmann and Braun, Elsag Bailey, and ABB. His main fields of interest are automation systems, especially DCS, and all aspects of computer validation. Schaefer holds a master degree in electrical engineering. He can be contacted by email: ernst-reiner.schaefer@abacon.org.

Abacon Gesellschaft für Mess-, Steuer- und Regeltechnik mbH, Landstrasse 81, 5430 Wettingen, Switzerland.
### Decision Criteria for Test Management Tools

Planning and deciding on the use of test management tools depends on a calculation – or at least – a rough estimation of costs and an assessment of the quality of testing and test documentation. Setting up a general one-size-fits-all calculation model for the Return On Investment (ROI) is impossible, as the formula depends on which of the many features of test management will be used, and which testing aspects have to be covered using conventional (manual) testing methods. However, the following aspects may be taken into account when deciding for a specific scenario.

![Test case life-cycle phases within a test management tool.](image)

#### Setup

<table>
<thead>
<tr>
<th>With Tool</th>
<th>w/o Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Significant efforts required to assess, implement, configure, and set up the tool (typically one-off effort). Users must be trained in the tool.</td>
<td>Lower efforts required, but storage location for test documentation needs to be defined.</td>
</tr>
</tbody>
</table>

#### Preparation

<table>
<thead>
<tr>
<th>With Tool</th>
<th>w/o Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tools typically provide a clear structure that needs to be followed. This kind of consistency may require more effort initially, but does often provide quality gains. However, bulk changes may not be supported or require more effort.</td>
<td>Rules for templates and for structured test specifications should be defined. Users are typically more experienced in using office applications.</td>
</tr>
<tr>
<td>Test assignments and compilation of test sets/sessions are typically well-supported by the tool. A tool provides easier communication of responsibilities. Tools are especially useful for distributed teams.</td>
<td>Needs to be resolved organizationally.</td>
</tr>
<tr>
<td>Typically, supported by the tool with minimal effort once versioning rules are defined.</td>
<td>Needs to be resolved organizationally (e.g., defined in an SOP).</td>
</tr>
</tbody>
</table>

#### Execution

<table>
<thead>
<tr>
<th>With Tool</th>
<th>w/o Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managing test execution and entering the results requires additional training. For trained users, entering test results and adding attachments is easier, as the tool provides guidance, thereby increasing quality and consistency. If the tool supports requirements, too, a traceability matrix can often be generated out of the box.</td>
<td>Finalizing the GMP-related test documentation, especially adding the appropriate attachments, is an error-prone activity that typically needs high effort. Creation of a traceability matrix requires effort and is an error-prone process.</td>
</tr>
<tr>
<td>Results of failed tests can easily be added. A direct link (and tracing) to the defect management system is typically provided (in some tools, defect management is already built-in). Repeated test execution is supported through tight integration of test and defect management.</td>
<td>Typically, defect management (i.e., documenting, tracking etc.) requires additional effort.</td>
</tr>
</tbody>
</table>

#### Test Status Management

<table>
<thead>
<tr>
<th>With Tool</th>
<th>w/o Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>The current status of all test activities can be visualized any time. Many tools come with extensive reporting tools (e.g., dashboard). Especially for long test phases, high volumes, and distributed teams, a tool offers significant benefits.</td>
<td>The test status has to be calculated manually.</td>
</tr>
</tbody>
</table>