

Agile Record

The Magazine for Agile Developers and Agile Testers



Distributed Team Management

www.agilerecord.com

free digital version

made in Germany

ISSN 2191-1320

August 2013

issue 15

How Scrum Changes Test Centers

by Klaus Haller & Konrad Schlude

Do you like agility? Do you like being celebrated like a rock star? There was once such an opportunity. You and the authors of this paper missed it. Until recently, developers lusted after tales of agile and Scrum projects. This changed as agile became the new normal for development projects and talks about Scrum mainstream. In the testing community, in contrast, Scrum brought uncertainty. Is agility the ultimate chance for developers to get rid of nasty testers? Or does agility only change job profiles? This article provides the answers.

Thesis 1: Agile Is About Development, Not About Testing and Testers.

“From niche to mainstream” – this sums up the trend towards agility and Scrum. Already 42 % of the participants in a 2009 international study report agile pilot projects in their organization; 9 % reported the majority or all projects are run agile [1]. In a Swiss study in 2012, 51 % of the participants state their company applies agile methodologies, mostly Scrum [2]. A German study from 2011 backs the dominance of Scrum. From the participants using agile methodologies, 57 % rely on Scrum, 27 % on internal methodologies. No other agile methodology reaches even 6 % [3].

Worldwide, the IT sector is moving to agility and Scrum. Thus, developers and managers must see benefits to these methodologies. Based on their exhaustive study, Vijayasathy and Turk summarize them in one sentence: “The ability to meet client needs and the delivery of quality software products on time are significant benefits of agile development” [4].

The impact on testers and test centers is not always clear. The *Agile Manifesto* [5] does not use the word “testing” even once. Also, in a Microsoft study about the benefits of agile software development, “increased quality” ranks only fifth (about half the votes of the topic ranked first) [6]. The study uses even with a broad definition of quality, from “test driven development” and “test automation” to “more stable feature sets”. It is an inconvenient truth for testers: agility is about development, not about testing or testers. Our advice for testers is simple: Accept the changes as farmers accept both good and bad weather. The new

world brings a broader spectrum of tasks for testers. Testers can now help improve software quality rather than merely tally its defects. This is a great opportunity.

Thesis 2: Scrum – What’s New?

Scrum is the most prominent agile methodology. Methodologies provide structures and best practices for projects to help them succeed. The best-known one in software development is the V-model. It has analysis, design, and implementation phases. Test phases follow: unit tests, unit integration tests, system tests, system integration tests, and user acceptance tests. In short, the V-model defines *when* to perform *which* test.

Scrum follows a different philosophy. It structures projects into sprints. Sprints have fixed durations, e.g. two or four weeks. At the start of each sprint, the Scrum team selects user stories. These are work packages which must be implemented at the end of the sprint. Further, the complete software must be stable and running again at the end of every sprint. This demands the same tests as the V-model, even if not specified explicitly. The test phases from the V-model become test aspects in Scrum. Projects decide *when* to schedule which tests with which priority. Figure 1 illustrates this. There is a linear order for test phases in the V-model. For Scrum, the distribution is more dynamic. Various test aspects are tested in a single sprint, plus the same aspects are tested again in later sprints.

Thesis 3: How Testing Differs If You Do Not Work for Microsoft or Google

Microsoft and Google offer software-based services such as Microsoft SkyDrive or Google Maps. They have large in-house development teams. They run all tests, from unit to system integration and user acceptance tests (Figure 2), though they sometimes call them something different. Their books inspire testers *how* to test [7] [8].

Most testers work in a different context. They work for small- and medium-sized enterprises (software vendors), developing custom or standard software for one or a limited number of customers.

Test Aspects		Temporal Order				
Test Typ	Validates	V-Model Project Progress →			Scrum Project Progress →	
Unit Test	Modules, classes, etc.	1				
Unit Integration Tests	Interaction between two modules etc.		2			
System Tests	Complete software			3		
System Integration Tests	Software coupled with other systems				4	
User Acceptance Tests	Requirements					5

Figure 1: Test aspects/aim and the temporal structure. The color intensity symbolizes the test intensity at a given point in time.

Qualification needed?



Foto: © iStockimages - fotolia.com

Sogeti Germany offers a wide range of qualified trainings in German!

iSQI CAT® Certified Agile Tester

TMap NEXT®

- Test Engineer (TMPTE)
- Test Manager (TMPM)

ISTQB® Certified Tester

- Foundation Level (CTFL)
- Advanced Level-Testmanager (CTAL-TM)

TPI NEXT® for Test Managers and Project Leaders

You can register until 30th September for any Sogeti training throughout 2013 in Germany and gain a free copy of the book „TMap NEXT® in Scrum“.*

*Paid participation in a seminar is required.

Please write the text „agile records“ on the registration form.



SOGETI

For more information please follow this link:
<http://www.sogeti.de/aktuelleseminare.html>



Saxonia Systems
 So geht Software.



WE ARE ALL ABOUT SOFTWARE.

TEST AND SEE US AT THE AGILE TESTING DAYS!

WWW.SAXSYS.DE

Other testers work for non-IT companies also called *customer IT organizations*. Mainly, they buy standard software and configure it to their company's needs. The customer IT organizations and smaller software vendors have different testing needs than larger IT companies. The dynamics of the customer-vendor relationship are major drivers, impacting the testing needs, especially in Scrum. The best way to illustrate the importance of dynamics is a comparison to drug development in the pharmaceutical industry. When developing new drugs, researchers work to minimize side effects. Societies could simply trust them to do so, but instead governmental agencies verify whether new drugs threaten patients' health. This is similar to developing good quality software and verifying its quality.

Software vendors test to find bugs and remove them. This helps developers improve code quality. To ensure quick feedback, Scrum emphasizes nightly builds. Overnight, all code is packaged and deployed on a test server. Then, automated regression tests run against the build. In the morning, when programmers come back, they know what works and what they must improve. Software vendors also test in order to manage project and liability risks. Testing is one way to document software quality. It helps in case of later customer disputes. Also, testing assesses the quality of the project and its progress. Managers can look at test reports and see which test cases ran successfully, which failed, and which could even not be started. This provides a better understanding of the project's progress, which is hard(er) to sugar-coat than simple status reports.

Customer IT departments test to guard production. Testers must prevent deployments which harm the IT and the daily work of users. They test to decide whether to accept a concrete software deployment. They do not test to improve software quality. Further, they test to verify whether the vendor fulfills the contract. Their focus is on user acceptance and system integration tests. For complex standard software such as ERP systems, they might do system tests to test the parameterization.

In-House Development	Software Vendor	Customer IT
Development → Deployment	Development → Shipping → Deployment	
Unit Tests	Unit Tests	
Unit Integration Tests	Unit Integration Tests	
System Tests	System Tests	System Tests
System Integration Tests	System Integration Tests	System Integration Tests
User Acceptance Tests	User Acceptance Tests	User Acceptance Tests
→ Improve Quality	→ Improve Quality	→ Guard to PROD
→ Manage Risks	→ Manage Risks	→ Contractual Verification
→ Guard to PROD		

Figure 2: Reasons to Test – In-House Development versus Vendor/Customer Relationship

A change from the V-model to Scrum impacts the need for the various tests (Table 1). Testing to “Improve quality” gains on relevance. Testers and developers work closer together and interact much more. Testing to manage contractual risks (vendor side) and contractual verification (customer side) becomes less important. Instead, customers give proactive input to the Scrum team. The need to test for project and reputational risks remains unchanged. Unchanged remains also the importance of tests guarding PROD from software which harms the daily work of the users.

IT organization type	Reason for testing			
	Improve Quality	Manage Risks	Contractual verification	Guard to PROD
In House (Test Center)	↑	Contracts: n/a Reputation: → Project: →	n/a	→
Software Vendor	↑	Contracts: ↓ Reputation: → Project: →	n/a	n/a
Customer IT Department	n/a	n/a	↓ (Continuous input!)	→

Table 1: Changing Priorities in Testing: From the V-Model to Scrum (↑ relevance rises, → relevance unchanged, ↓ relevance lower)

Company size and strategy influence whether non-IT companies develop software internally or whether they outsource. This fourth thesis addresses the latter ones, *lean customer IT departments* with outsourced development, and their need for testing and testers. These IT departments manage suppliers to ensure that the IT empowers the business. They need, besides project managers, three types of specialists: business analysts, integration experts, and testers. Business analysts define the requirements to ensure that the IT supports the business as good as possible. Integration specialists install the software in-house. They ensure it collaborates with satellite systems. Testers verify the software quality.

V-model and Scrum projects need them at different times in the project. The V-model has one peak in the beginning. Here, business analysts prepare the requirements (Figure 3). Then, the software vendor takes over. He develops and tests up to the system test level. Then, the customer IT departments get involved again. This is the second staffing peak towards the end of the project. The software is deployed and integrated into the customer IT landscape; system integration and user acceptance tests follow. Mostly, there is only time to fix small bugs. Thus, these tests remind more on teachers grading an exam. The two peaks in the V-model challenge smaller IT departments. They need complete, ready-to-work teams for a limited time. Outsourcing is the only solution. IT service providers and consultant companies can help.

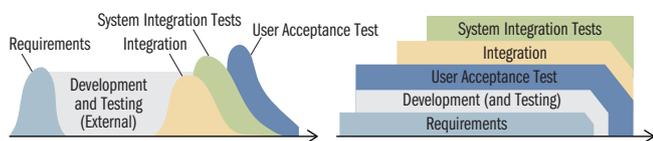


Figure 3: Schematic illustration workload and tasks for Lean IT (left: V Model, right: Scrum)

Scrum needs the same skills as the V-model. However, Scrum balances the resource needs more over the complete project time. Requirements engineering and testing takes place throughout the complete project, though there might be a testing peak at the end. This enables customer IT departments to keep such skills (partially) in house or hire contractors. This is bad news for some test service companies. Just sending large teams of junior testers to customers does not work anymore. Scrum

projects need testers a broad skill-set, domain-expertise, exploratory testing experience, and even business analysis skills.

Thesis 5: Testers in Agile Projects – from Autistic Testers to Communication Champions

Testers have in-depth know-how of testing methodologies. They are trained to analyze requirements to derive test cases. They aim to find defects, so they raise questions no one focused on new software development raises. Testers stand out for checking the software from various perspectives. Thus, they bridge the gap between development and business with their end-to-end view.

The main task for testers for V-model and Scrum projects is clearly test preparation and execution. In Scrum projects, they take over additional responsibilities. Since testers and developers work at the same time (and not in different phases), they can work more closely together. Testers support developers also with the following tasks: reviewing user stories; checking for testability; crossfunctional activities, such as test automation (beyond unit tests) and coordination in a distributed team; and demo sprint review – when the new integrated software package is presented to the product owner, a tester can execute test cases related to acceptance criteria.

Testers take over tasks and may be integrated into development teams. This requires the ability to work cooperatively. V-model testers manually execute clearly defined test cases. They work with precision and focus over a long period of time, even if the work is repetitive. This fits, for example, with SAP’s idea to hire autistic testers [9]. In agile projects, however, testers need good communication skills to really foster collaboration with developers.

Thesis 6: Agile Projects need Testing on a Program Level

In Scrum, testers act less as “guards” to production. There is also no need “to guide the software into production” as testers do in the V-model. In Scrum, the product owner takes this role (though testers can certainly become future product owners). The testers’ strength of overview is no longer required in development teams.

While this old responsibility of testers vanishes, new challenges arise in large projects and programs embedding several projects. In a program, the product owners’ efforts increase, resulting in several product owners. The ideal agile world reaches its limits. When various project teams and product owners focus on their deliverables, they may find nothing works when plugged together. Here the testers’ overview is once again needed. The solution is system integration test on the program level by a testing team combining application know-how and a deep understanding of the overall system.

The optimal organization depends on whether system tests and system integration tests are done by the same organization. If so, testers will work in a normal sprint as system testers. From time to time, there are system integration sprints (Figure 4). In system integration sprints, testers from the various development teams come together and form a temporary team

for holistic tests. If system and system integration tests are done by different organizations, there is a need for a separate system integration test team, requiring other tasks outside system integration sprints.

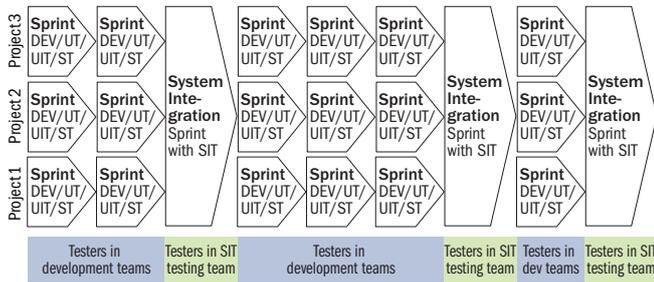


Figure 4: Testing for Programs

Thesis 7: Realign IT Departments to the Strategic Needs of Agility

Successful organizations align the goals of employees and the goals of their managers. If a manager’s performance evaluation depends on a software roll-out date, no tester reporting to this manager should be expected to stop the rollout for quality reasons (see [10] for a broader discussion). In general, three rules are important for organizing testing:

- If a tester verifies software quality, he must not report to the software development project manager.
- If a tester lists and evaluates the risks of a project, he should not report to the software development project manager.
- If a tester and developer work together, they should be placed in the same team.

We apply these simple rules for the V-model and Scrum for three company types: software vendors, lean IT companies with no internal development, and mixed IT companies with internal and outsourced development.

For the V-model, independent test centers are perfect for software vendors (Figure 5, left). Software development teams and testing teams are separated but on the same hierarchical

level. This fosters quality. When the vendor changes to Scrum, they need to reorganize their testing. First, testers have to be embedded into agile development teams. They can move completely to development project teams or they can come from a tester pool. A tester pool remains in a QA organization and offers career paths in testing. Testers in the development teams test everything from unit tests to system tests. Second, if different development teams work on related projects, a program manager must coordinate them. The manager relies on a system integration test team (see Thesis 6) to ensure that the development teams deliver interoperable software. Third, when testers are spread over the complete organization, IT organization must enforce company-wide testing standards. A small team for defining standard processes and reviewing projects helps.

As a simplification, we assume a lean IT organization to have an operations unit and a project unit (Figure 6). The project unit coordinates external suppliers developing custom software or parameterizing standard software. The operations unit runs an internal data center or manages an IT service provider. The operations unit requires an internal or outsourced testing team for system integration tests. It is a stable team guarding production. Thus, it belongs to the operations unit, never to the project unit.

When companies run V-model projects, they need testers (and business analysts) during peak times in the project (see Figure 3). Larger teams might be needed to start working immediately. Only external providers can provide such services.

When the project unit changes to Scrum, staffing needs also change. The workload gets more stable. Testers work more exploratorily. Their roles might even fuse with the business analyst role. Outsourcing is not “peak sourcing”, i.e. bringing in a large team of testers for some days, weeks, or months. Outsourcing turns into long-term “know-how sourcing” relationships between lean IT organizations and suppliers with testing, business analysis, and domain expertise.

Mixed IT organizations have internal software development plus outsourced custom software development or use standard software. In the V-model, a classic test center incorporating all testers is the right organizational form.

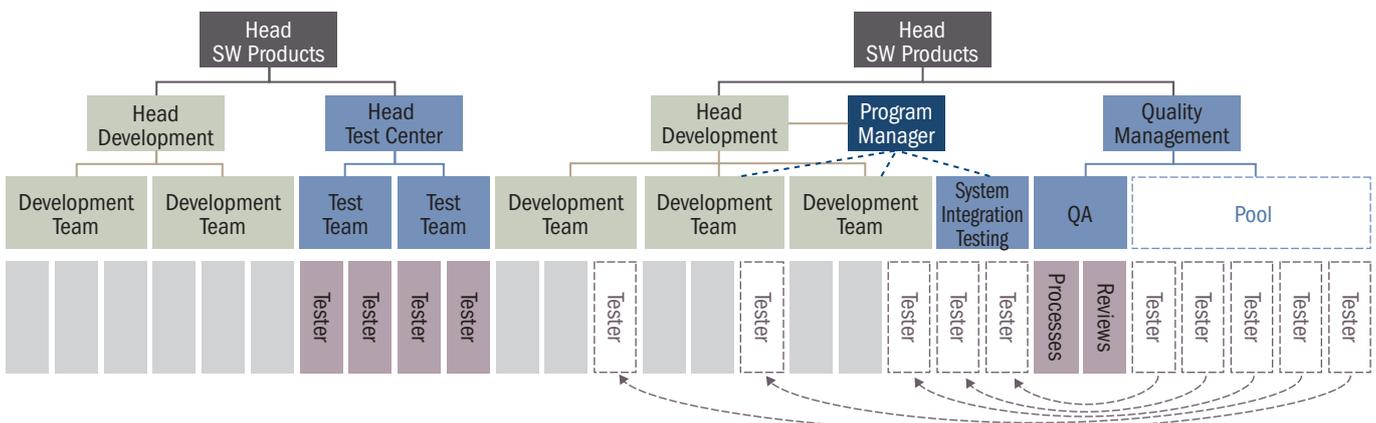


Figure 5: Organizational Patterns for Development Companies – Test Center Style vs. Agility Support

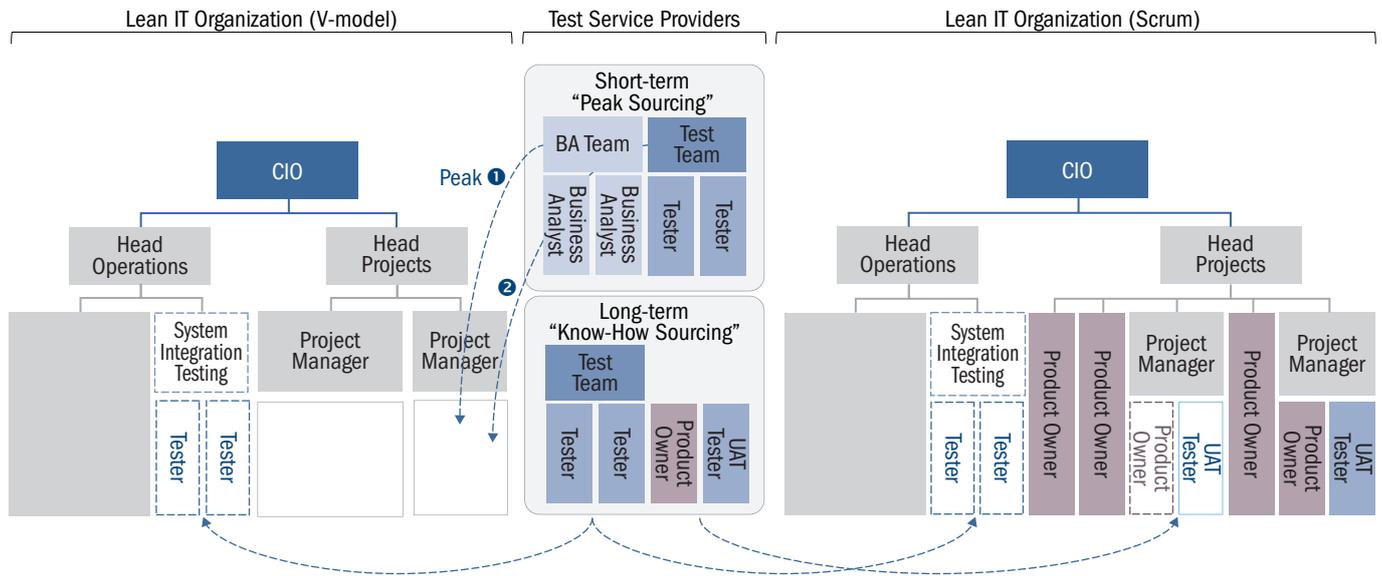
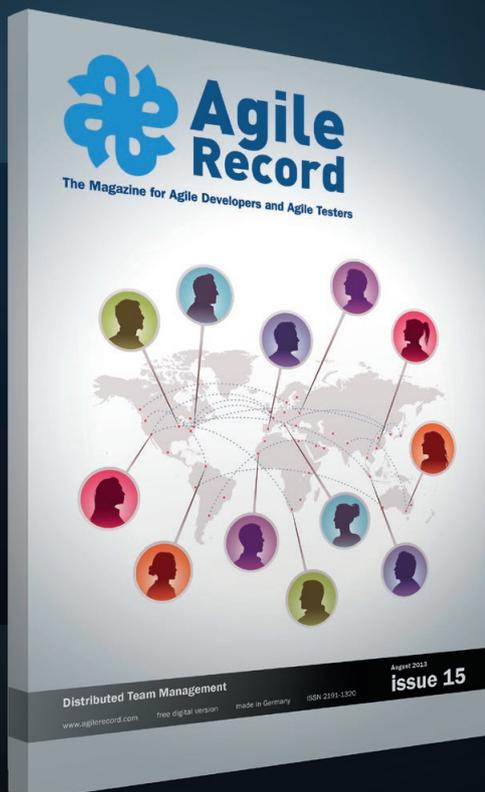


Figure 6: Organizational and Sourcing Patterns for Lean IT Companies – V-model vs. Agility

With Scrum, testers are split into two main groups: agile testers and guards. Agile testers work for development teams. They help with everything from unit tests to system tests. Agile testers can come from a tester pool in the test center or have development managers as line managers. A quality assurance

team in a test center must ensure that guidelines are defined and followed even if the testers are spread across development teams. Besides agile testers, there are guard testers. They focus on system integration and user acceptance tests. They ensure that the new software never harms production.

Show your full potential!



Call for Articles

Become an author for the Agile Record magazine and share your knowledge and experience with other professionals from the field.

The next issue of Agile Record, on the topic of “Refactoring”, will be out in November 2013.

Get your articles in for review by September 15, 2013!

Find more information on our website at:

www.agilerecord.com

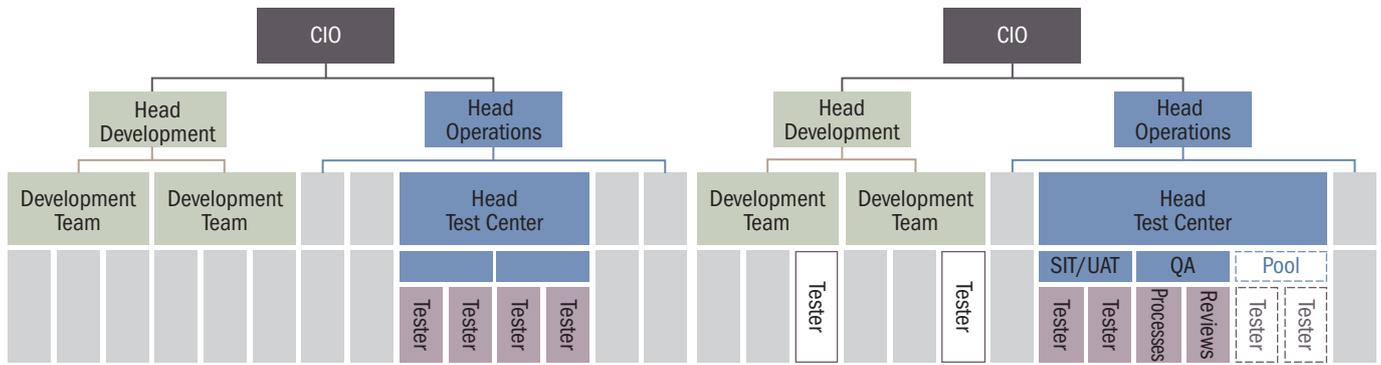


Figure 7: Organizational Patterns for Mixed IT Companies – V-model (left) vs. Agility (right)

Conclusion

Is agility the ultimate opportunity for developers to get rid of nasty testers? No, it is not. Will Scrum change test organizations? Yes, we expect major changes. First of all, the clear temporal order of test phases is replaced with more dynamic planning with Scrum. Further, we expect three groups of testers. First, guardian testers in IT operations will run system integration tests to protect from any harm to the IT landscape. Second, agile testers embedded into development teams will help to raise software quality. This is important for software vendors and internal software development. The role of these testers might even merge with business analyst roles and therefore demands good communication skills. Third, larger development projects will see testers on the program level. They ensure that the software developed by the various agile development teams fits together. Test service provider will have to redefine their offerings for the three new areas of testing. As test centers change, so too the roles of testers. If you see your colleagues struggling with the uncertainty, you might share the words of Ovid: “Fortune and love favor the brave.”

References

- [1] Borland Agile Testing Survey Results 2009 – Update. White Paper, July 2009.
- [2] SwissQ Agile Trends & Benchmarks Schweiz 2012: Wo stehen wir – wohin geht es? http://www.mcpa.biz/wp-content/uploads/2012/07/SwissQ_Agile_Trends_2012.pdf
- [3] P. Haberl et al. *Umfrage 2011 – Softwaretest in der Praxis*. dpunkt.Verlag: Heidelberg 2012.
- [4] L. R. Vijayarathy, D. Turk. “Agile Software Development: A Survey of Early Adopters.” *Journal of Information Technology Management*, Volume XIX, Number 2. 2008.
- [5] K. Beck et al. *Manifesto for Agile Software Development*. <http://agilemanifesto.org/>

- [6] A. Begel, N. Nagappan. *Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study*. 1st International Symposium on Empirical Software Engineering and Measurement (ESEM '07). September 20th and 21st, 2007, Madrid, Spain
- [7] A. Page et al. *How We Test Software at Microsoft*. Microsoft Press: Redmond, WA, 2008.
- [8] J. Whittaker et al. *How Google Tests Software*. Addison-Wesley Longman: Amsterdam, The Netherlands, 2012.
- [9] “SAP to Hire Autistic Staff in Foundation’s First Global Alliance,” Bloomberg: May 22, 2013. <http://www.bloomberg.com/news/print/2013-05-22/sap-to-hire-autistic-staff-in-foundation-s-first-global-alliance.html>
- [10] J. McKay. *Managing the Test People – A Guide to Practical Technical Management*. Rocky Nock: Santa Barbara, CA, 2007. ■

> about the authors

Klaus Haller



Klaus Haller (www.klaushaller.net) has worked for more than eight years in the consulting industry, primarily in the banking sector. His focus is testing information systems landscapes, including methodology, processes, and organization, especially in the areas of test data management, compliance testing, and IT risk. Klaus publishes regularly about testing and is a frequent speaker at conferences. He has worked for Swisscom since 2006.

Konrad Schlude



Konrad Schlude has worked in software testing since 2004, mainly in the finance industry. He is an experienced test manager with experience in several areas from test automation, offshoring, and agile testing in large development programs.