VDI/VDE-Gesellschaft
Mess- und Automatisierungstechnik

Testing of Networked Systems for Industrie 4.0

Abstract

Digitalization in industrial automation and the associated changes in product development and industry has far reaching consequences. The purpose of this report is to detail the effects of digitalization on Industrie 4.0 Components and Systems and to raise awareness among research and industry stakeholders of new challenges as they reach the limits of their current testing processes due to the complexity of their products, manufacturing facilities, and their IT environments. Additionally, it provides an overview of existing standards and norms in the information technology sector.

By means of use cases, these changes are showcased and the results are summarized and attached in a tabular format. In the conclusion, a detailed explanation of the individual sectors and components affected by Industrie 4.0 is provided.

In summary, the results of this status report are:

- To manage increasing complexity, a division of the considered systems into subsystems is necessary. This subdivision requires the use of well-defined interfaces.
- Proprietary third-party solutions typically do not allow complete testing by the user. In this case, certified execution of standardized tests can provide the necessary confidence in the solution.
- Selected proven standards from other domains could be adopted in automation technology after adjustments are made to them.
- Quality standards requirements for current software will also be required for software produced in batch sizes of 1 in the future.
- Challenges represent themselves in the test setup and the simulation environment.
- To obtain efficient and reproducible testing in complex and changing environments, model-based techniques in test automation will become increasingly relevant.
- Metrics increase in relevance to estimate sufficient coverage.

Düsseldorf, April 2018

Prof. Dr.-Ing. Michael Weyrich
Chairman of the Technical Committee
„Testen vernetzter Systeme in Industrie 4.0“

Contents

1 Preamble

It is well known that the complexity of networked components and their development is increasing. As such "increasing complexity" appears as a common theme through various sections of this status report. Therefore, this theme and its definition are discussed here. We have the following understanding of complexity [1]:

> The complexity of a system derives from the whole of its interdependent features and elements, which form a holistic relationship. In Industrie 4.0, an increase in the complexity of systems is evident, among other things, as a result of these factors:
>
> - heterogeneity of the components
> - interaction of components within the system
> - cross-system communication
> - modification of system structures through ad hoc networking and reconfiguration
> - scaling of the system
> - joint consideration of software, hardware and communication.

Compared to traditional, non-networked systems, Industrie 4.0 Systems contain many, if not all, of these factors. The growing demands on these systems influence these factors. For example, future production facilities should react with much more flexibility to product changes through ad hoc networking and reconfiguration. In addition, it is desirable to maintain traceability of a product over its entire life cycle. Industrie 4.0 Systems are internally and externally networked: the components within the system interact. Cross-system, cross-domain and machine-to-machine (M2M) communication takes place. In addition to testing the system functions, it becomes necessary to consider the influence of the modular system on such cooperating systems and vice versa. This increases the test effort.

There are already approaches to manage the complexity of Industrie 4.0 Systems or Internet of Things (IoT) systems. For this purpose, several reference architectures have been developed in recent years. These include the Industrial Internet Reference Architecture (IIRA) of the Industrial Internet Consortium (IIC) and the Reference Architecture Model Industrie 4.0 (RAMI 4.0) of the German Electrical and Electronic Manufacturers' Association (ZVEI). In this case, RAMI 4.0 includes the management shell and enables many aspects to be encapsulated. Nevertheless, only the complexity can be reduced by these reference architectures; they do not describe systematic test processes.

Increasing complexity of the systems results in an increased complexity of the test cases. Furthermore, more test cases are needed to achieve similar test coverage with additional considerations to consider. New standards, norms and testing procedures are needed to handle changes to Industrie 4.0 Systems.

Test scenarios in such systems consist of the system test as well as the component test with knowledge of the internal (white box perspective) and only of the external (black box perspective). To meet these challenges, existing test processes can be adapted or new test processes can be developed. The same applies to test languages to achieve consistency in the development of Industrie 4.0 Systems. Additionally, metrics to measure complexity and test coverage must be adapted to new scenarios to quantify both test effort and success.

The following section describes the changes caused by Industrie 4.0 in various application areas by means of use cases. Following a presentation of the results, an explanation of the most important terms is provided.

2 Use cases

Considering the increase in complexity of Industrie 4.0 systems, this chapter describes the expected changes in specific areas of automation technology. These are clearly presented by experts of the respective domain via use cases.

2.1 Test of process control systems

2.1.1 Current procedure

Process control systems (PCS) are divided into subsystems during the test. Subsystem protection includes, for example, interoperability testing for the integration of smart field devices into the control system or batch system testing. Additionally, the entire system will be tested during the factory acceptance test (FAT) at the manufacturer and at the site acceptance test (SAT) in the plant. All tests are held to a scheduled time and defined test regulations. (e.g. DIN EN 62381).

2.1.2 New challenges

Challenges arise from different application scenarios and from the use of process control systems from different manufacturers or different process control systems from the same manufacturer. This complexity is dealt with by exact test specifications for the individual (sub-)systems of the manufacturers. Furthermore, testers require a high level of know-how and precise knowledge of the system to be tested. For special cases, the knowledge of the responsible specialists is required.

2.1.3 Future use case

Considering the increase in vertical system integration, e.g. as a prerequisite for emerging value creation networks, the need for communication increases over the individual levels of the automation pyramid. Dependencies between the process control system (PCS), production execution system (PES) and production planning (enterprise resource planning, ERP) are becoming increasingly critical. A test of only the PCS is therefore no longer sufficient. It is also necessary to test the communication and behavior of the PCS for events from these communications. Typical questions for future use cases include:

- What data will be exchanged?
- Which systems expects which reactions?
- What happens in the case of communication failure or faulty communication?
- How is the communication handeled in a secured manner (security)?

Additional challenges arise from different organizational responsibilities: Typically, ERP and the PCS are located in different departments, for example, IT or automation technology.

One solution strategy offers the standardization of test processes and test case descriptions. This concerns both the individual systems (PCS, PES, ERP) and their interfaces, as well as the contents of the communication. It is necessary to test each system interface for typical external faults. For this, the dependencies between the systems must be clearly defined. The systems themselves could then undergo additional testing in a robust manner.

2.2 Field devices (component test)

2.2.1 Current procedure

Each product development is specified twice according to a defined process. The requirement specification answers the question of “What is required?” and the target specification clarifies the nature of the realization. According to the V-Model, the components of the acceptance test (release) are defined approximately simultaneously. This is the responsibility of a separate, independent department. All properties described in the target specification are tested **against the requirement specification**.

Furthermore, integration tests are carried out. For these tests, the product is tested alongside other associated components (e.g. PLC). The goal is to discover and avoid errors or weaknesses in the implementation of third-party components.

2.2.2 New challenges

Almost every communication-enabled device today is too complex to be tested regarding all conceivable input and output data. The state graph of such a system would become almost infinitely complex. Functional testing focuses on the associated specification.

Therefore, an attempt is made to thoroughly test all relevant requirements of the specification. This requires a systematic approach in terms of test generation and high automation in terms of test execution.

Stress tests are used in an attempt to simulate complex overload situations and to stabilize them with suitable countermeasures. Dealing with errors in integration testing is challenging: Not every third-party device is available for testing, and a workaround for compatibility issues is not always desirable. In contrast to the procedure of the final approval test, the product development of large product families today leads to a continuous improvement of the product family (continuous integration). The error category expectation of user robustness is becoming increasingly important. If used incorrectly, the user usually compromises the product and its robustness or the intuitiveness of its operation.

2.2.3 Future use case

Rapid intervention in the software, such as adding functions or reusing libraries, creates difficulty in verifying the overall system. The resulting complexity will become increasingly problematic in the future. First, testing should be completely automated to enable repetition of tests immediately after system changes. Based on this, test generation methods (e.g. model-based testing) could systematize and automate test generation (see Figure 1).

In addition, a reduction in component complexity is necessary. Input data for components must be kept to a minimum. Switching off unused functions reduces potential attack surfaces (see IEC 62443). Penetration tests should be carried out in such a way that potential backdoors do not cause any additional attack vectors. Today, static code analysis or human review are already very effective methods to find such vulnerabilities. The remaining interfaces must be simplified again to allow full testing. For this, tests should be broken down into modules or functions (units).

Since the user cannot be required to test a library in detail, a kind of warranty for the reuse of software functions is necessary. This confirms the successful completion of a comprehensive test. The biggest challenge in the future will be whether a function is comprehensibly structured and specified. In this case, no absolute statements are possible because, as is the case with penetration tests, dependencies on the person performing testing are unavoidable.

2.3 Software production for standard applications

2.3.1 Current procedure

The production of standard software for standard applications in plant and machine automation today follows a classic, waterfall development model. As a result, the main activities of testing arise at the end of the linear arrangement of development activities. Since the generated code (programming languages in accordance with DIN EN 61131-3) can only be executed in conjunction with a controller, the significance of the concluding system test also increases. Using virtual commissioning (VIBN) methods, these system tests can also be carried out largely in the laboratory and/or at the workplace of the test engineer. The test strategy used provides a requirement-based test case selection for the dynamic test. The test documentation is based on a company-specific tailoring of IEEE 829.

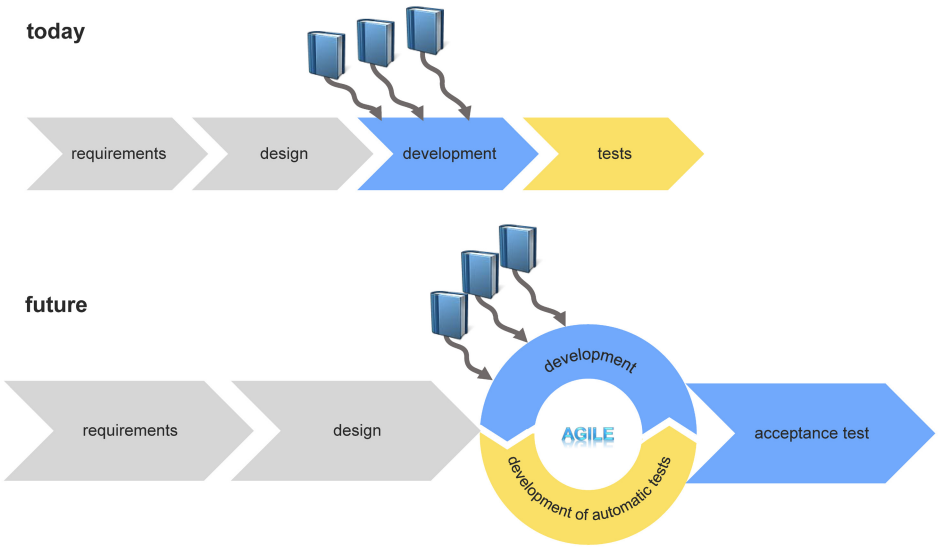

Figure 1. Change of test cycles in the development cycle of a component

2.3.2 New challenges

Focusing on upper test levels in conjunction with a selected development model leads to a relatively long error retention period (period between mistaken action and detection of the error effect). The remedy here is to introduce an iterative development model and shortening of the development cycles in conjunction with a parallelization of the test cycles.

In addition, the concept of phase restriction is introduced. By applying static test methods and introducing additional test levels, deviations in the phase of the development process are discovered and remedied as they arise. The monitoring of metrics to evaluate the test process enables continuous improvement of test and development processes.

2.3.3 Future use case

Due to increasing networking, it is becoming apparent that both the amount and complexity of software is increasing and its significance in the quality of the overall product is steadily increasing (systems are becoming multi-systems). Additionally, software development is evolving from a more scientific process to a production process (industrialization of software development). This includes the production philosophy that accompanies Industry 4.0. The quality requirements, which are already known for standard software, will also be required for software produced in batch sizes of 1 in the future. It can be expected that testing, which used to focus on the (standard) product, is moving towards a standardized process with high automation of test creation and test execution. The increase in test process knowledge shifts the focus towards a holistic quality assurance strategy that covers the entire product lifecycle (see Figure 2). In this case, in addition to the testing, error prevention becomes a priority. The basic idea that the quality of the software is significantly influenced by the quality of the creation process is already reflected in various (test) process reference models.

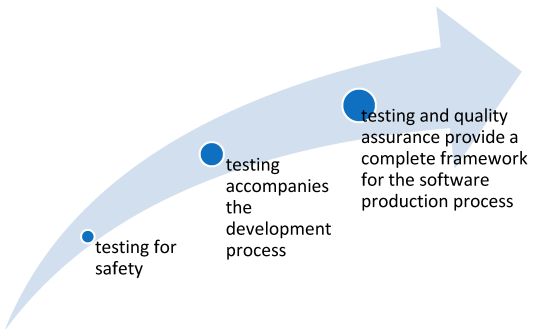

Figure 2. Development stages of software testing

2.4 Automotive functions in IoT

2.4.1 Current procedure

Currently, test methods in the automotive industry are primarily geared toward electronic control units (ECUs): The classic system under test (SuT) is an ECU embedded in a largely static network topology. The system limits are the bus frames and the I/O interfaces. These tests operate on bus signals, diagnostic protocols and, at times, bus frames.

2.4.2 New challenges

ECUs have not been simple control blocks for quite some time. They feature complex algorithms and data models, communication interfaces from the PC environment, asynchronous programming paradigms, and service-oriented architectures or adaptive AUTOSAR. Simulation and configuration effort increases significantly and a direct assignment of specific functions to individual ECUs is becoming increasingly difficult. As a result, it is necessary to perform more testing at the application level to emphasize functions and to abstract communication technologies into a meaningful framework.

2.4.3 Future use case

In the future, testing of vehicle functions will be virtual and highly automated, and will particularly target their interaction with dynamic ad hoc networks (IoT).

Challenges reside in the modeling of the test setup and the simulation environment, e.g. in the identification of suitable domain models, and in the definition of suitable interfaces for test cases. In addition, choosing the appropriate level of abstraction is difficult. For example, the accuracy of the simulation of behavior, time or communication, as well as consideration for countless subscriber configurations must be considered while testing with limited resources.

First steps in this direction suggest that test and modeling languages must be increasingly optimized and expanded in the future, especially regarding the depiction of dynamic processes. Also, the benefits of modern programming paradigms must be made available faster and more usable in the context of testing vehicle functions.

2.5 Security

2.5.1 Current procedure

In the manufacturing and process industry, networking of automation equipment for controlling and monitoring production is becoming increasingly common. The network interface is an important attack surface for threats such as computer viruses and other malware. There are numerous factors that cause a large variety of operating system versions and applications to be used simultaneously. As part of asset management, a target-performance comparison through network scans would be helpful. In combination with a vulnerability scanner, known vulnerabilities of operating systems and applications could also be identified automatically.

2.5.2 Challenges

The longevity of network-enabled systems and the lower patch frequency make it difficult to assume that any system in an industrial network will withstand a network or vulnerability scan without malfunction. This complicates testing with security tools that are required for reliable operation in other areas. Due to the threat of losses in the event of unplanned shutdowns of equipment and the risk of unrecoverable or difficult to repair errors due to testing for previously unknown safety problems during the lifetime of the system, testing while in use is only possible in rare cases.

2.5.3 Future use case

Due to progressive networking, the risk of security vulnerabilities in the operating phase, which could be exploited via the network, increases. However, these cannot be detected by tests, since many of the installed components were not designed for networked use. As a result, the risk of downtime or malfunction lies with the operator. The following trends can be observed:

- When purchasing new equipment, security requirements play an increasingly large role that may be accompanied by acceptance tests.
- Test standards are developed for the testing of industrial network components, e.g. from the Federal Office for Security in Information Technology.
- Increased segmentation of networks makes it possible to operate testable components in smaller, less sensitive subnets, e.g. through virtual local area networks (VLANs).

2.6 Summary of use cases

In the described applications from different domains, certain aspects can be found. Table 1 illustrates the dependencies among them.

Considering the use cases mentioned above, the terms shown are particularly relevant. Therefore, they are explicitly considered and explained in the following section.

Table 1. Mapping of definitions with use cases

	Agile Methods	Interfaces	Standards for Describing Test Cases	Test Automation	Model-based testing	Metrics
Test of process control systems		x	x	x		
Field devices (test components)	x	x	x	x		
Software production for standard applications	x		x	x		x
Vehicle functions the context of IoT		x	x	x	x	
Security		x	x			

3 Terms and definitions

3.1 Agile methods

3.1.1 Introduction

Numerous procedural models and methods have emerged for structuring development projects. As agile methods become increasingly important in product development, they are explicitly addressed. Among other reasons, this can be attributed to the greater integration of the customer into the development process. It specifically does not include any evaluation of established development methods that continue to be justified.

Above all, the term agility denotes speed and flexibility in implementation. These are important attributes in the fast-paced IT world. Thus, agile methods are finding growing use in businesses. One of their main goals is to manage the increasing complexity of development. In the future, this will be the biggest challenge facing companies, according to an IBM global CEO study. In addition, shorter lifecycles, rapid technology change, increasing competition and globalization of markets have a major impact on product development. Innovation projects, such as Industrie 4.0, are characterized by a multitude of incomplete information, uncertainty, and interdisciplinary cooperation. Therefore, a rigid structure of the development process is often not effective.

The above points illustrate the importance of agile methods in today's products. In these processes, special attention is paid to continuous interaction with the customer. Frequent coordination occurs within the project team and testing takes place during development. This agile testing is essential for complex projects. Here, the primary focus is on automated testing, and will be discussed in more detail below.

3.1.2 Importance

Agile processes, such as SCRUM, require automated testing methods. Otherwise, the required quality of a software product cannot be guaranteed. Additionally, a core feature of testing is to ensure the reproducibility of the tests. This typically is not possible without automated distributed systems such as those found in Industrie 4.0.

Automated testing of distributed systems is also very challenging. This is because of the large number of independently operating components. The required effort continues to increase if the components are developed by different manufacturers and have a decentralized organization. Synchronization of test processes due to the large number of largely self-sufficient subsystems is very complex.

3.1.3 Explanation

The definition of uniform test processes and test contents is useful to cope with a higher degree of modularity. This results from the need for test automation and from the large number of components used with a variety of different manufacturers. To ensure a uniform base, cooperation between hardware and software manufacturers and system integrators is necessary.

3.2 Interfaces

3.2.1 Introduction

The overall increase in complexity in Industrie 4.0 Environments requires countermeasures to keep the cost of designing, building and operating such systems manageable in the future. One approach to reducing complexity is the greater division of systems into subsystems that interact via interfaces. A distinction must be made between at least three different layers of interfaces:

- **process layer** (What information is transferred for which purpose?)
- **software layer** (What is the structure of the transferred information?)
- **hardware layer** (What is the physical method for transferring the information?)

The interoperability of the subsystems depends on the precise definition and error-free implementation of these interfaces on all layers. There are also at least three different dimensions in terms of interface testing:

- **white-box perspective** (What is internal representation of the interface?)
- **black-box perspective** (What is external representation of the interface?)
- **system-perspective** (How does the interface behave in the context of the whole system?)

As a result, testing of interfaces or of complete systems composed of subsystems is a very real, multi-dimensional challenge.

3.2.2 Importance

In the following example, the challenges of interfaces in the Industrie 4.0 context are outlined at the software level:

The assumption that software is fundamentally flawed is already widespread today. Although incorrect in its absoluteness, it highlights an important point: the error rate depends significantly on the complexity of a system. A reversal of this trend from today's comparatively simple systems to upcoming, highly complex Industrie 4.0 Systems is not very optimistic.

The reduction of complexity usually stands no chance in the market. On the other hand, a more promising approach is the breakdown of a complex system into smaller, less complex subsystems. As a result, each of these subsystems can be realized with little effort, and only when these subsystems are (re-)integrated into the overall system do complexity and the resulting effort potentially become problematic once again. This problem can be solved with well-defined interfaces.

If the definition of an interface is known, individual components can be completely tested using this interface definition. In this way, basic interoperability can be ensured and the risk of system integration is significantly reduced. For the purposes of reusability and division of labor (parallel development), the complete definition of interfaces is of central importance: only the interface definition is used for this and the implementation of other components need not be considered or known.

This approach can also be used for testing. If the definition of an interface is defined as machine-readable (for example, as part of the programming language), then the functional behavior can be automatically tested, including the influence of resources. The breakdown of this procedure to the functional level can be defined as module tests or unit tests.

3.2.3 Explanation

In addition to testing all interface layers, tests in all interface dimensions are especially important. It is true that every single dimension is necessary, but only all tests together are sufficient for a conclusive and comprehensive evaluation of the entire system. Additionally, the individual dimensions that must be tested by different actors must be considered: As a rule, only the manufacturer can carry out a white-box test and only the customer can carry out a system test.

Therefore, it is necessary to examine interface tests in their multi-dimensionality and complexity and develop general (meta) criteria for conducting them to create a basis for using Industrie 4.0 Components via transparency and standardization of tests. However, it is impossible to create specific test instructions, as they will differ from one application to another.

In terms of system testing, reasonable limits must be defined. With increasing connectivity, testing of the entire system becomes impossible due to its complexity and variability.

Furthermore, it is important to investigate technical concepts in the implementation of interfaces, such as data diodes and gateways. In addition, extensions of common programming languages may be necessary, such as machine-readable interface definitions for static code analysis and for the (additional) automatic testing of small units. This enables control over the ever-finer division of complex systems into individual components and to ensure the functionality of components in the network.

3.3 Standards for test case descriptions

3.3.1 Introduction

The standard ISO/IEC/IEEE 829 (or its successor ISO/IEC/IEEE 29119-3) has become well established for software testing. This standard defines different types of documents to capture different test aspects. Of these, the test design, test case and test procedure specification are relevant to the description of test cases. By using these types of documents, a uniform, traceable and reproducible specification and implementation can be ensured.

Due to growing complexity, increased networking and the high degree of functionalities implemented by software in Industrie 4.0, a need for standardized test descriptions is expected in the future for this domain. Since the ISO/IEC/IEEE 829 standard primarily specifies structuring and rough content of the different document types, the question arises as to whether these stipulations are sufficient and suitable for test cases in the Industrie 4.0 environment. A few years ago, requirements were identified for description of test cases in the field of telecommunications that went beyond the requirements of ISO/IEC/IEEE 829. Therefore, the European Telecommunications Standards Institute (ETSI) has developed advanced standards for the telecommunications sector. Using the already established standards for test case descriptions

for Industrie 4.0 seems to be advantageous due to existing tools and the associated expertise.

3.3.2 Importance

The use of standardized test description languages has shown the following advantages in the field of telecommunications and are also considered important in Industrie 4.0:

- support simplified reviews
- uniform s means for the description of test cases
- ensuring interoperability between testing tools

The following ETSI standards can be utilized for the description of test cases in the field of Industrie 4.0:

- Test Description Language (TDL) [6] for an abstract description of a test case
- Testing and Test Control Notation Version 3 (TTCN-3) [7] for the specification of test cases to be executed

TDL is used for specifying test descriptions and presenting test execution results. TDL bridges a methodological gap between abstract test requirements and executable test scripts. A test description written in TDL consists of the test configuration, test description and test data components. The test description can be designed either graphically or by text. A major advantage of TDL is its expandability to other domains, such as Industrie 4.0. Additionally, ETSI provides an open source reference implementation of TDL.

The Test and Test Control Notation Version 3 (TTCN-3) is a description language for the specification of test cases for the purpose of testing communication protocols and services. In this language, APIs of software components can also be addressed. Due to its ability to simultaneously drive and monitor multiple test points, the use of TTCN-3 for Industrie 4.0 testing seems appropriate. In addition to purely functional tests, TTCN-3 can also perform interoperability, robustness and performance tests. TTCN-3 is already established today in test automation in the field of communications technology.

3.3.3 Explanation

Although the two standards TDL and TTCN-3 seem to be suitable for specifying test cases for Industrie 4.0 at different abstraction levels at first glance, further analysis is essential. Domain-specific requirements must also be checked that may require adjustments.

3.4 Test automation

3.4.1 Introduction

A test process can be divided into the activities of test planning and control, test analysis and design, test realization and execution, and test evaluation and reporting (see fundamental test process in Figure 3).

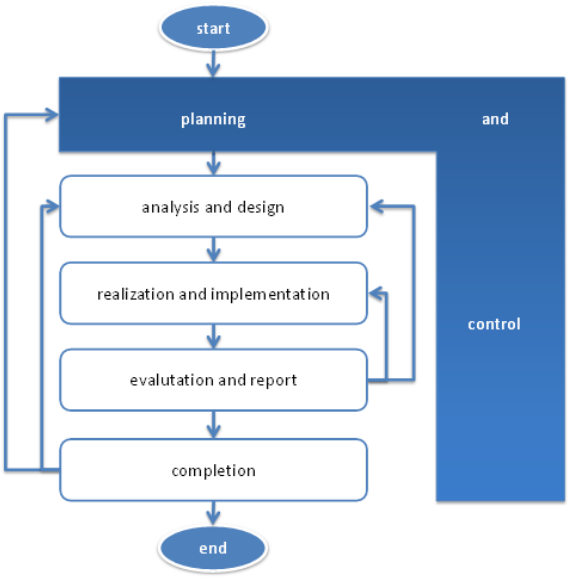

Figure 3. Fundamental test process in accordance with ISTQB®

Test automation involves all activities within a test process. In practice, however, the automation of the activities of test realization and implementation as well as test evaluation and reporting is currently most popular. There are efforts to automate test generation within the activity of test planning by methods of (model-based) test generation.

Test automation offers efficiency advantages when many similar test cases or identical test cases need to be routinely executed after software modifications (regression tests). In addition, the flexibility of a development process through test automation is greatly increased, as fast, low-effort and systematic feedback to implemented changes takes place. Thus, the automation of testing is an indispensable part of agile development.

In practice, test automation systems are usually implemented individually. In addition to TTCN-3, it lacks standards. Thus, an exchange of test data between tools from different manufacturers is currently difficult to achieve.

3.4.2 Importance

Test automation will play a fundamental role in providing quality systems and features in the future. To ensure proper functionality, they need a comprehensive, systematic and above all automated test. Considering the anticipated high reconfiguration rate, continuous testing is indispensable. The motivation to automate results stems from the complexity of future systems, which require many test cases to ensure quality. Both the creation of these test cases and the execution of the test can only be realized by high-level automation with reasonable effort. For efficient and reproducible performance of tests in complex and changing environments, model-based techniques in test automation will become increasingly relevant.

3.4.3 Explanation

The challenge for future test systems for use with Industrie 4.0 applications or Functions is their high-grade networking. Test systems must be able to control this network and control it if necessary. Today's classic black box testing approach is still up to date, but unsuitable for testing highly networked systems. Important test functions such as the identification of faulty components and subsystems, targeted error indications in distributed systems, etc. are not feasible with these approaches both in the test description and specification as well as in the test execution.

3.5 Model-based testing

3.5.1 Introduction

Model-based testing was originally introduced to automatically generate test cases from specification models. A specification model contains the required behavior of a test object, which is derived from the requirements independently of the implementation. Algorithms then automatically generate systematic test data and associated test sequences (Figure 4). Academic and commercial tools already exist for this purpose. Common modeling notations are graph-based models such as state machines, activity diagrams, petri nets, etc. However, text-based modeling languages are also used as the input model of a test generator.

A specification model must be correct in relation to the requirements, otherwise no correct test cases can be derived. The necessary verification of this is a non-trivial process. Manual testing (reviews, modeling patterns, etc.), as well as simulations and automated formal verification can be used with tools for model checking and/or automatic proofing.

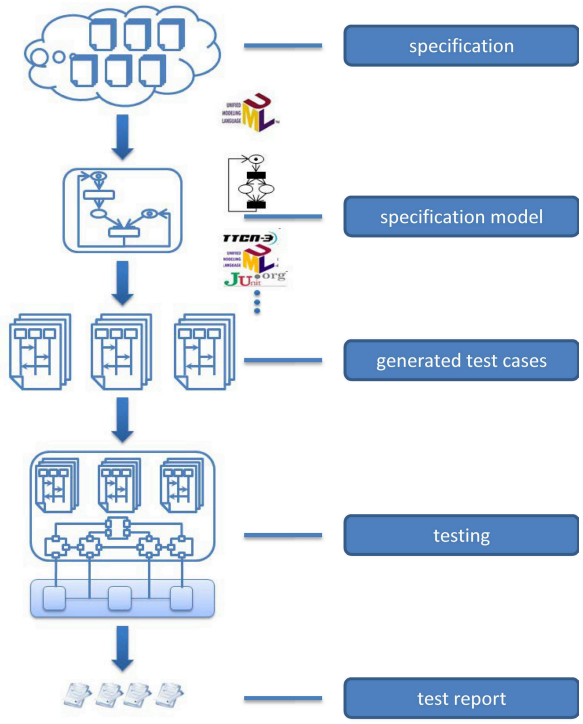

Figure 4. Model-based test process

Further developments show that other methods are used for model-based testing. It is possible to use models within a test process (modeling of the test architecture, the test environment, the test data but also the test cases, see UML Test Profile [10]). Since modeling focuses on the essential aspects, the creation and documentation of test activities can be simplified. Virtual commissioning is also used in conjunction with models to test factory automation equipment in timely manner. Primarily, PLC programs and/or robot controllers are tested in this case.

3.5.2 Importance

Model-based test methods have a high potential to efficiently create test processes with little effort.

However, very high demands are placed on the test engineer. Manual modeling can quickly become very large and complex, representing an initial hurdle for the entire model-based testing process.

3.5.3 Explanation

Currently, the method of model-based testing is not fully implemented in practice. This is primarily due to the high entry barrier of modeling the specification model, which often must be derived from text-based requirements. Here, it would make sense to support test engineers or modelers to considerably simplify the modeling step. For this, research efforts attempt to

Table 2. Savings potential via model-based testing [5]

Study	Time Cost without MBT	Time Cost with MBT	Savings via MBT
Ericsson/Conformiq	20 h per test case	5.5 h per test case	73 %
Siemens/Trapeze	2.67 h per test case	0.67 h per test case	75 %
Sepp.med	2.04 h per test case	1.36 h per test case	43 %
Microsoft	2.37 days per issue	1.38 days per issue	42 %
Conformiq/Forrester	6.396.565 $ total	1.288.794 $ total	30 % test specification 84 % maintenance

automatically transfer natural language specified requirements into suitable models (natural processing-language).

3.6 Metrics

3.6.1 Introduction

Systems and processes in Industrie 4.0 are often safety-critical and exceedingly complex. This prioritizes the testing of the tools and software used to prove certain quality and reliability criteria. Among other things, metrics can be utilized to assess product quality. The challenge here is to meaningfully combine suitable metrics in such a way that the quality of a system or process can be assessed based on a few key figures. This requires new guidelines that identify appropriate metrics and methods for Industrie 4.0.

3.6.2 Importance

Ensuring the interoperability of systems despite increasing interconnectivity is a major challenge. Therefore, testing and verification of individual components is crucial. Metrics offer the opportunity to formally describe quality and reliability requirements and to demonstrate the quality of a product in a comprehensible way. It makes sense to identify or redevelop suitable metrics to make testing in Industrie 4.0 Systematic and comprehensible.

3.6.3 Explanation

Metrics are already extensively used in software development and are also used in controlling software projects for assessing product quality. If and how existing concepts can be transferred into Industrie 4.0 must be examined. To identify potentially suitable software development metrics for use in Industrie 4.0, the following provides a brief overview of common test methods and metrics in the IT space.

Test methods that can be used to generate metrics include functional test cases based on requirements, I/O and communication testing between different components, and performance testing. They regard the system and test as a black box. A frequently used metric is the requirements coverage. It measures how many of the formally specified requirements are covered by test cases. Considering additional successful and error-free test cases of a requirement, requirement conformity of a product can be determined. With the help of the requirement conformity, it is possible to assess the quality of the product, since a low conformity is very likely to be accompanied by a low product quality.

In addition to the requirements-based tests, there are also several tests that are derived directly from the source code and thus treat the SuT as a white box. These are based on the implementation. There are different criteria for the derivation of the test cases:

- every line – line coverage
- all statements – c0
- all branches and all exits – c1
- all memory accesses – d0, etc.

These criteria are used directly as a metric for the test coverage, so they are also called code coverage. However, they allow only conclusions about the correct implementation and make no statements whether the examined software is error-free a condition that might need to be communicated clearly. Whether the code also meets the requirements cannot be estimated using this method. This requires the above-mentioned functional tests.

Another important group of test metrics comes from the field of probabilistic testing. The basis of all probabilistic test methods is the knowledge that complete

freedom from errors in software is usually neither required nor proven. Instead, it is sufficient to prove a certain rarity of failure in a particular operating environment. For this purpose, statistical methods are used which look at the SuT from a black-box or system perspective. A measure of reliability in this case is the probability of failure. Further characteristics according to IEC 60050 are, for example, the mean time to failure (MTTF), which indicates the average operating time until failure of a component, or the average time between failures (MTBF). The disadvantage of all these parameters is their statistical origin. To determine them, there must already be many empirical values with the respective components or a corresponding number of tests (several thousand) must be carried out. This quickly gets uneconomical, especially for small batches.

This list of different metrics is not exhaustive and is intended to illustrate that there are already many metrics and key figures from the IT area. A current challenge is identifying and, if necessary, adapting suitable software metrics for use in Industrie 4.0. In terms of hardware metrics, such a process should not be necessary as they should not differ significantly. An important aspect in which both areas differ concerns interoperability between components. Evaluating the interoperability of a system in Industrie 4.0 is difficult to achieve using common metrics. A test against standardized interfaces is expected to become significantly more important. New metrics may be required for such interface or interoperability testing, as well as for assessing security issues, which are specifically tailored to the Industrie 4.0 application.

Bibliography

[1] http://wirtschaftslexikon.gabler.de/Definition/komplexitaet.html

[2] SEW Industrie 4.0 White Paper

[3] IBM Global CEO Study: The enterprise of the future

[4] Tilo Linz: Testen von SCRUM-Projekten, dpunkt-Verlag

[5] „Modellbasiertes Testen: Hype Oder Realität?“ Stephan Weißleder, Baris Güldali, Michael Mlynarski, Arne-Michael Törsel, David Faragó, Florian Prester, Mario Winter, ObjektSpektrum 06/2011

[6] http://www.etsi.org/technologies-clusters/technologies/test-description-language

[7] http://www.etsi.org/technologies-clusters/technologies/testing/ttcn-3

[8] ETSI: Testing and Test Control Notation Version 3 (TTCN-3), www.ttcn-3.org

[9] Pretschner, A.: Zum modellbasierten funktionalen Test reaktiver Systeme, Dissertation, TU München, München, 2003

[10] Object Management Group: UML Testing Profile, http://www.omg.org/technology/documents/formal/test_profile.htm/, 2007

Technische Regeln

ANSI/IEEE 829:2008 Standard for Software Test Documentation

DIN EN 61131-3:2013-04 Automatisierungssysteme in der verfahrenstechnischen Industrie, Werksabnahme (FAT), Abnahme der installierten Anlage (SAT) und Integrationstest (SIT)

DIN EN 62443 IT-Sicherheit für industrielle Automatisierungssysteme (IEC 62443)

DIN IEC 60050 Internationales Elektrotechnisches Wörterbuch

IEEE 829:2008 Standard for software and system test documentation

ISO/IEC/IEEE 29119-3:2013-09 Software-und Systemengineering; Software-Test; (Teil 3: Testdokumentation)

The VDI

Spokespersons, Designers, Network Engineers

The fascination for technology drives us forward: For 160 years, the VDI Association of German Engineers has provided vital impetus for new technologies and technical solutions for improved quality of life, a better environment and greater standard of living. With around 150,000 personal members, the VDI is the largest technical and scientific association in Germany. As a voice for engineers and technology, we actively help shape the future. Each year, more than 12,000 volunteer experts process the latest findings to promote our technology base. As the third largest issuer of technical regulations, the VDI is a partner for the German economy and science.

More VDI Publications for free download

www.vdi.de/publikationen

VDI Status Report „Industrie 4.0 – Technical Assets – Basic terminology concepts, life cycles and administration models“

VDI-Statusreport „Industrie 4.0 – Begriffe/Terms“

VDI Status Report „Industrie 4.0 Components – Modeling Examples“

VDI Status Report „Industrie 4.0 Service Architecture – Basic concepts for interoperability”

VDI Status Report „Oppertunities with big data – Best practice“

VDI Status Report „Oppertunities with big data – Use cases“

VDI Status Report „Reference Architeccture Model Industrie 4.0 (RAMI4.0)“

VDI Status Report „New Business Models for Industrie 4.0“

VDI – The Association of German Engineers
VDI/VDE Society Measurement and
Automatic Control (GMA)
Dr.-Ing. Thomas Sowa
Phone +49 211 6214-223
sowa@vdi.de
www.vdi.de/gma