

**We asked three software developers for test automation for their opinions about test automation, and share their thoughts and expertise here with you.**

#### THE FOLLOWING TELEMOTIVE TEST AUTOMATION TEAM MEMBERS WERE INTERVIEWED:

**Bastian Raymann** is the Group Lead and the Product Owner of the software solution Telemotive Test Automation.

**Florian Kempter** is the Team Lead of Telemotive Test Automation Development.

**Rolf Druegh** is the Telemotive Test Automation Core Developer and the Proxy Product Owner Core.

#### 1. What is test automation and what types of tests can you automate?

**F. K.:** First let me explain what testing means. Before you launch a product on the market, it has to be tested to see if it works as desired. It is worth considering whether automation is suitable for this. Test automation is a supporting but not a standalone tool to manage the number of tests required for each product and to improve the tests themselves. There are two types of tests. You can automate simple and time-consuming tests. Stress testing combines both: when you test your product under extreme conditions in different environments over a longer period of time.

**B. R.:** Automation is a process which performs repetitive tasks with minimum employee intervention to reduce human effort. If a test engineer has to repeat a job more than once, then instead of doing the same process again and again we create a piece of software that does the job to reduce human labor.

Test automation is the process of testing software or applications using an automation testing tool, e.g. on a computer, to find the defects. In this process, test scripts are executed and results are generated automatically by our tool Telemotive Test Automation.

**R. D.:** Test automation basically means that an automatic machine/computer is doing a test that is otherwise done by a tester (person). Generally, every test can be automated, but some types of tests are better candidates.

#### 2. When is test automation required?

**F. K.:** Test automation would be profitable for complex products that require high test coverage, let's say at least 500 test cases. It is also required for those product features that have to run under continuous stress tests.

**B. R.:** When you want to reduce time and monetary resources spent on test cases in the medium and long term. When test engineers have to run a huge amount of test cases anywhere and anytime. When manual testing is impossible due to the complexity of your product features. When you have to test on several different hardware or software platforms and configurations.

**R. D.:** A typical use case are regression tests. Regression tests have to be done very often and/or on a regular schedule (e.g. for every software/firmware release or even in case of changes). Automating those tests saves time and money as the development can focus on the actual implementation since even smaller changes can be tested automatically.





### 3. Is it possible to achieve 100 % automation? Why not?

**F. K.:** There is no product that can be tested 100 % using automation because not all functionalities can be tested, for example product design, user-friendliness, and good layout.

**B. R.:** 100 % automation would be difficult to achieve because there would be many edge test cases and some cases that are rarely executed. Automating these rarely executed cases will not add value to the automated suite.

**R. D.:** It is not always possible to automate all tests because a testing tool needs a technical interface for every test step. Sometimes this interface is not available. From our experience, it means more effort if such an interface has to be implemented later during the product development process. It is cheaper if the testing interface is considered in the design phase, because then it is easier to optimize its implementation.

### 4. What are the primary features of a good automation tool?

**F. K.:** In the first place, it must be easy to use and give a fast result when testing a basic function of a product. A high level of configuration must be possible, as tests can be very versatile and you cannot accurately forecast what you have to test since the same product, for example an infotainment system, can have very different functions. You must also be able to adapt and change the automated tests yourself. If the tool has features that fit my needs, then it is a good tool.



**B. R.:** Some good features would be:

- Test environment support and ease of use
- Independence of environment and system
- Good debugging facility, e.g. automatically created rich log files
- Robust object identification for HMI outputs
- Object and image testing abilities
- Object identification

**R. D.:** We think that a good automation tool should be easily extendable and support rapid prototyping even for complex tasks. It should also be possible to automate tasks of the tool itself, so other tools can use the automation tool and integrate its functionality.

### 5. What are the steps involved in the automation process? What do you have to consider in the planning of an automation project?

**F. K.:**

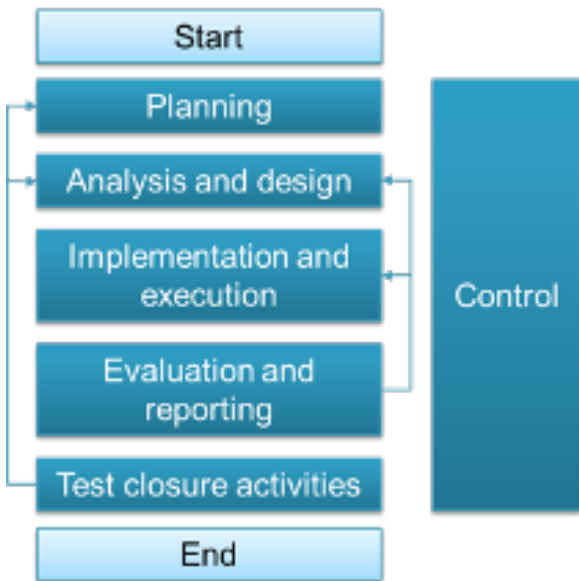
1. You have to decide whether the product can be tested automatically.
2. Can I run the test cases in a simplified/intuitive way? Is it possible to automate time-consuming tests in particular?
3. What kind of errors do I expect in the application? What might errors look like?
4. Consider use cases: Where is the product used and how? What are the product's possible applications? What does the environment where the product is used look like? Which features are required for this?

**B. R.:** When you plan an automation process, I gladly refer to the following model: the testing process of ISTQB. Testing is a process rather than a single activity. This process starts with test planning, then continues with designing test cases, preparing for execution, and evaluating status, and ends with test closure. So we can divide the activities within the fundamental test process into the following basic steps:

- 1) Planning
- 2) Analysis and design
- 3) Implementation and execution
- 4) Evaluating exit criteria and reporting
- 5) Test closure activities

Test control has the following major tasks:

- measure and analyze the results of reviews and testing,
- monitor and document progress, test coverage, and exit criteria,
- provide information on testing,
- initiate corrective actions,
- make decisions.



**R. D.:** When we start a project, we analyze the tests and group them based on efficiency and the testing interfaces required. The former means checking a) whether tests can be easily automated (less effort) and b) if there are benefits to development. Of course, if most of the interfaces required are available, this means less effort and has a positive effect on the efficiency rating. Regarding benefit, we consider that if a test has to be done very often (like regression tests) and/or it consumes a lot of the tester's resources, then its benefit rating rises.

## 6. Can you describe the framework used in Telemotive Test Automation?

**F. K.:** The TTA tool is a framework itself. With this tool, based on Python 3, developers can use open-source packages and benefit from elaborate reporting with clear visualization. TTA can be connected to almost any hardware, and thus facilitates the testing of embedded systems. Due to its open-source approach, tools for testing web interfaces and backends can be connected easily. REST APIs can be used to connect not only the software under test in the development process but also any existing software tools.

**R. D.:** We based our automation tool TTA on Python, because it is widely used (universities and institutes) and there are already a lot of modules that are useful for testing – like modules for performing complex calculations, image recognition, OCR, getting data from measurement devices, providing interfaces to databases, and different tracking systems (defect/issue management), etc. Furthermore, TTA allows the user to access all of its functions in Python natively, which means that it can be automated itself (automatic start of test cases, test case sequences, analysis and processing of results, etc.).

## 7. How many test cases have you automated per day since working remotely?

**F. K.:** Independent from working in an office or remotely, we can automate a huge amount of test cases. At the moment, we are testing five products and easily running 500 test cases each day. This makes at least 7,000 test cases in total.

**B. R.:** Thanks to our keyword-driven framework, we can automate unlimited test cases by using implemented keywords. Our engineering teams are able to set up project-specific keyword development quickly.

**R. D.:** I did not automate tests myself but supported testing teams by implementing and extending our TTA test automation tool.

