PIKETEC
software engineering

codebeamer & TPT

Whitepaper

Whitepaper codebeamer ALM & TPT

Abstract

This white paper gives insights into the integration of codebeamer and the testing tool TPT. Against the backdrop of this integration the importance of the traceability of requirements and test cases is illustrated.

In particular, an introduction to codebeamer and the testing tool TPT is given. Moreover, the importance of requirements-based testing is stressed and the development process throughout the release cycles with codebeamer and TPT is described.

Introduction

Throughout the development process the engineer is facing a myriad of tasks, two of which play a central role. Requirements need to be implemented and test cases need to be created and executed.

Moreover, running test cases in order to verify that implementations match the requirements is also done for quality assurance purposes.

Lastly, the implementation of all requirements needs to be checked against the corresponding hardware integration and vehicle integration level.

Since safety norms (e.g. ISO 26262) demand traceability of requirements, an integration of all of the above steps is not only of organizational but also of legal importance.

This white paper illustrates the test process using codebeamer and the testing tool TPT

Introduction codebeamer

codebeamer is an Engineering and Application Lifecycle Management (EALM) platform for advanced product and software development. It was
designed to help create better products in a faster and more efficient way through its unique set of features and integrations. codebeamer offers requirements management, development management, quality assurance and testing, Agile project management, DevOps, product line and risk management, powerful review features, regulatory compliance management and release management – all in an integrated, highly efficient, and user-friendly platform.

Introduction testing tool TPT

The testing tool TPT is a tool for testing embedded control software. It is used for the automatic creation, generation and execution of test cases. Moreover, all tests can be assessed and managed requirements-based.

All development phases from unit and module testing over software integration testing up to vehicle testing can be performed with TPT. All test environments are supported from MiL, SiL, PiL to HiL and automated driving tests.

The testing tool is also suitable for testing safety critical systems (ISO26262).

Relation of artefacts towards one another

Generally, test requirements describe the “shall-be behavior” of a system and the tests itself verify whether the system does what it should. Therefore, linking requirements and tests is key, and requirements-based testing plays a central role in testing with TPT. Besides, implementations are also linked to requirements but that is not the issue at stake here.

Essentially, what you want to show is that you have tested sufficiently in light of the requirements. In order to prove the completeness of your testing efforts, you’ll have to link requirements with the corresponding tests. As a result, untested requirements can be easily identified and reported.

Development through multiple release cycles

In order for the development process to run smoothly, every step of the way needs to be facilitated and supported by all means possible. This means, first of all, using an ALM tool such as codebeamer to organize releases and to manage the traceability of artefacts such as tests and requirements.

In the development process there are changes in requirements over different development cycles and releases. With TPT, changes between releases and their effect on testing can be easily tracked. This way, it is very easy to determine which requirements have changed and which test cases are affected by a new release cycle. This simplifies the process significantly.

Integrating ALM into TPT comes in quite handy when directly importing requirements and tests. You can also first develop the tests in TPT, then export them into codebeamer. In other words, the integration allows for many automated setups and proves to be very flexible in use.

Eventually, this enables a cost and time-efficient progression of development and release cycles.

The essential steps to along the automated test process are:

1. Import requirements from codebeamer into TPT
2. Import tests from codebeamer into TPT (If you already specified tests)
3. Import linking of artefacts imported in 1. and 2.
4. If applicable, create new tests in TPT or adapt existing tests and export them back to codebeamer
5. Implement tests in TPT
6. Perform tests in TPT
7. Export test results as test runs into codebeamer for further processing

In codebeamer ALM requirements are structured as shown below:

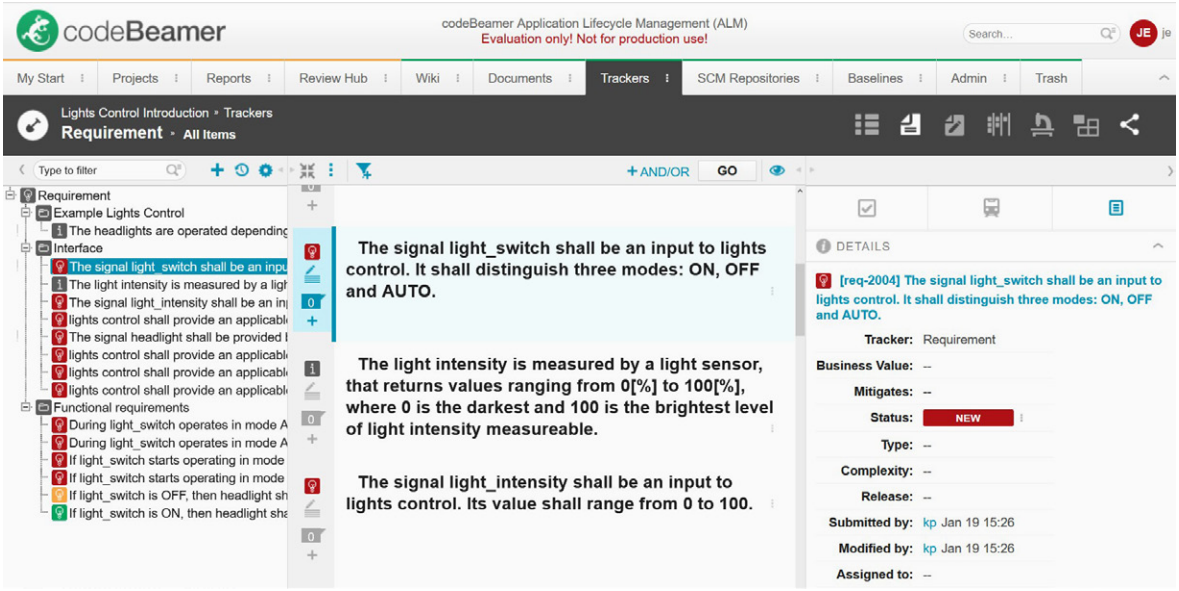

Figure 1 Requirements View in codebeamer

Requirements can be imported from requirements trackers or management platforms such as an ALM tool into TPT. Imported requirements in TPT for the given example are show in the figure below.

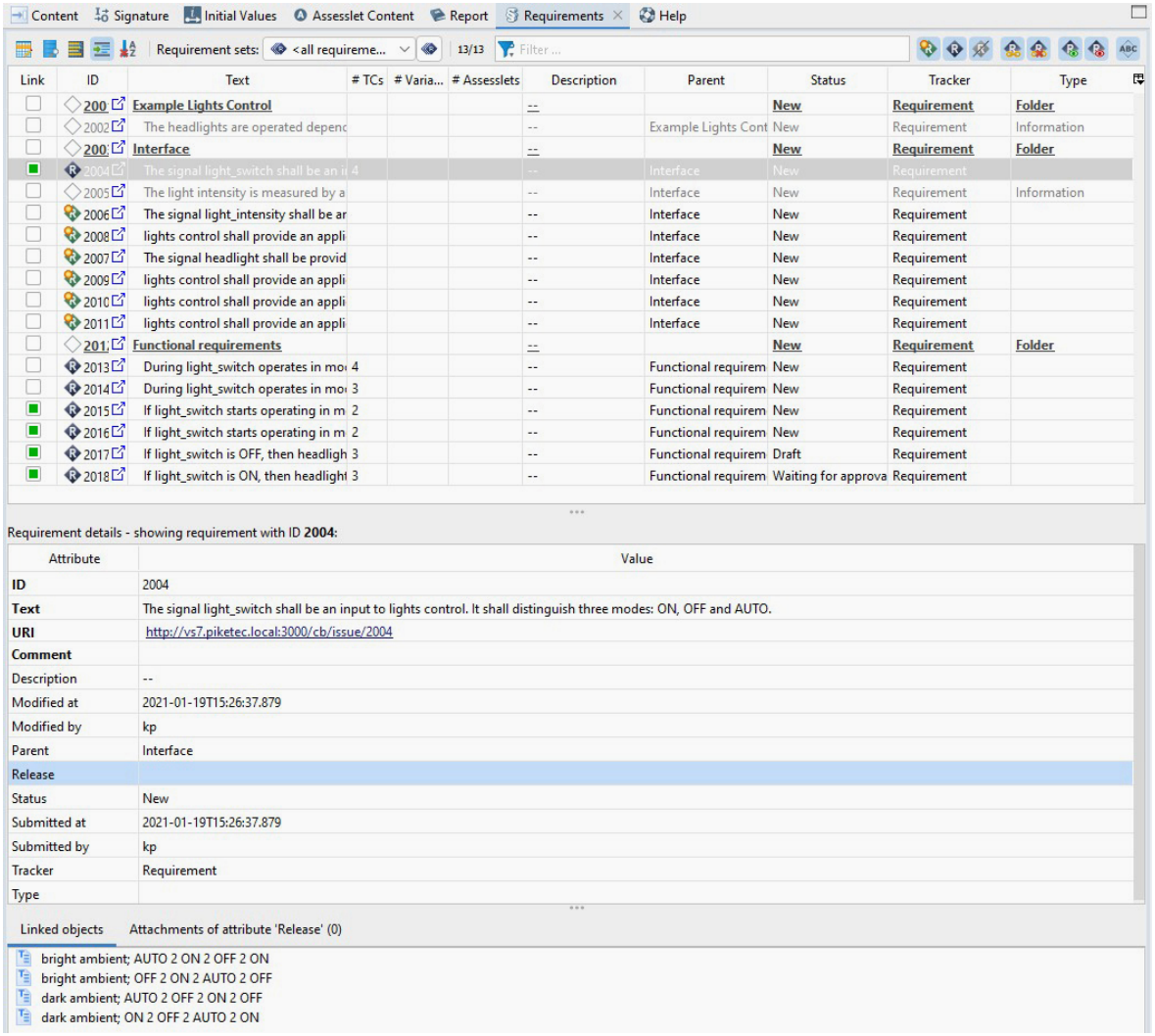

Figure 2 Requirements View in TPT

Once all requirements have been imported, tests can be implemented. If there are already existing test cases in the ALM tool, TPT will create and link the test cases accordingly (maintaining traceability). Next, the test implementation can then be done in TPT. New test cases that are created in TPT or existing ones that are modified can be exported to codebeamer.

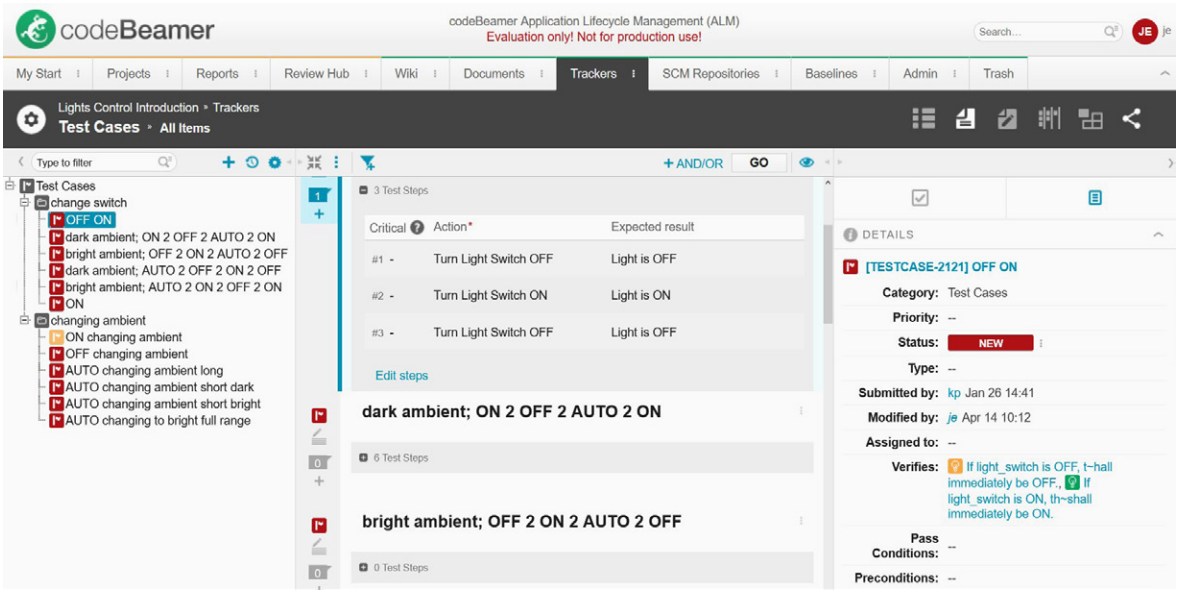

Figure 3 Test Cases in codebeamer

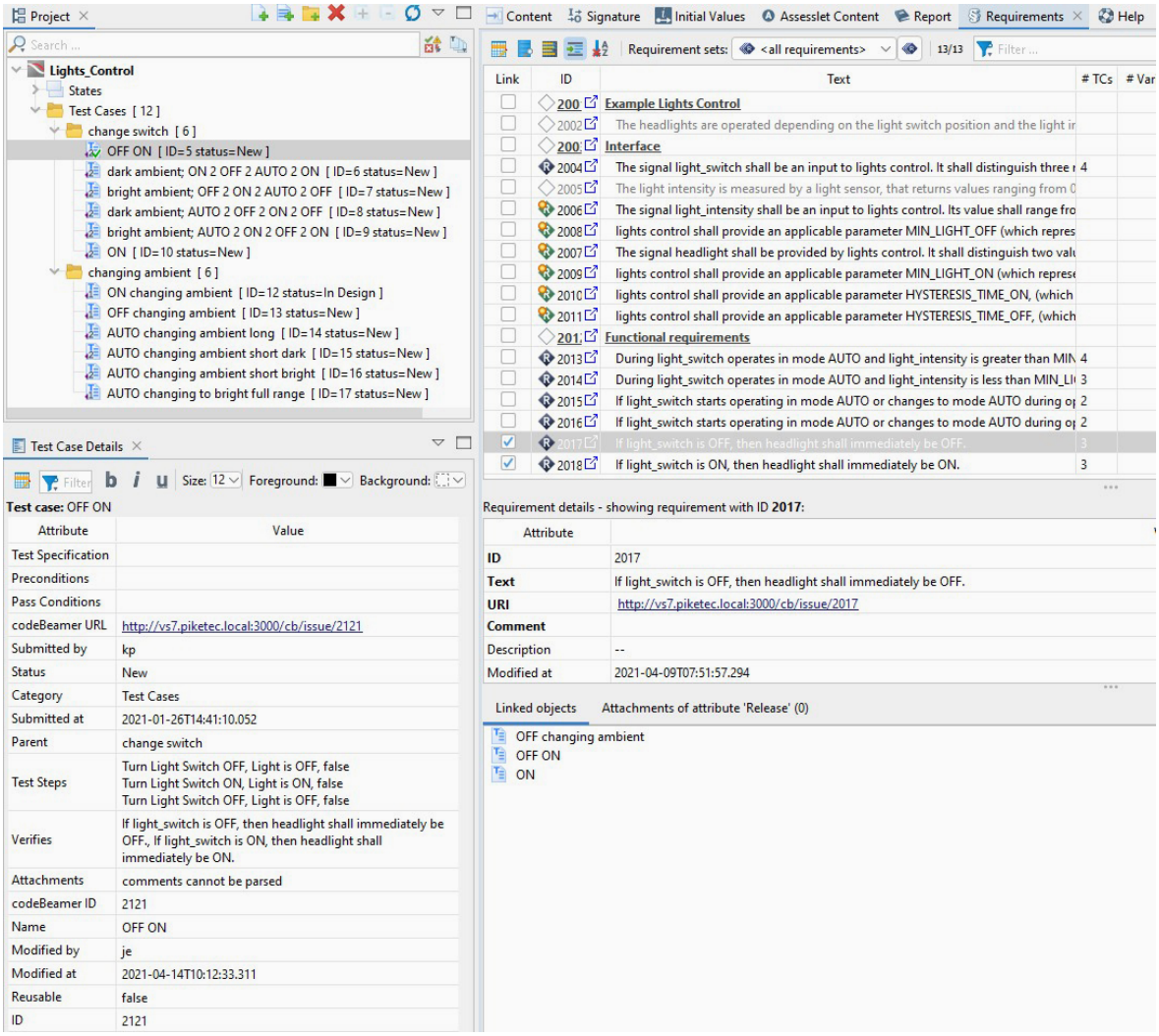

Figure 4 Imported Test Cases and Linked Requirements in TPT

In codebeamer you can categorize your test results, for example, per model or integration level. codebeamer also provides an overview over the entire project including all artefacts and their status, thereby considerably facilitating test project management.

After test cases have been executed, the results can be exported to codebeamer. Results are maintained in the Test Run tracker, as shown in the figure below.

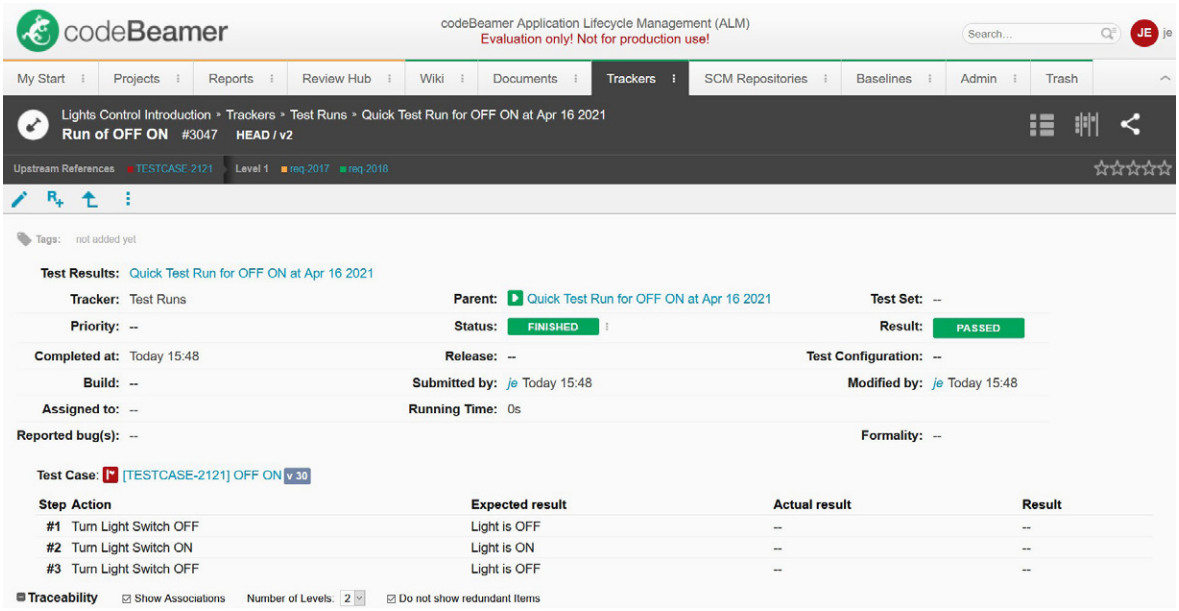

Figure 5 Test Run View for Test Case after Results Export to codebeamer

Throughout all of the described phases artefacts are linked to one another across both tools, maintaining constant co-traceability.

Release management

For new releases one would import changed requirements into TPT. TPT then highlights all changed requirements since the last release and highlights all artefacts in TPT that are related to the changed requirement. All test cases that are affected by a changed requirement are highlighted, as well. Changes in requirements can also be easily tracked. This makes working on new releases more efficient.

Summary

This white paper highlights the integration of codebeamer and the testing tool TPT, describing the development and requirements-based testing with both tools throughout the release cycles.

First, codebeamer and the testing tool TPT are briefly introduced. Second, the relation of the artefacts, as well as the consistent exchange of information between both tools in the testing process is outlined, stressing the central role of requirements-based testing.

In the following, the development process through multiple release cycles is described. This section lays out the project management, test management and legal requirements of developing safety critical software, identifying codebeamer and the testing tool TPT as particularly suitable full round trip support tools thanks to their integration.