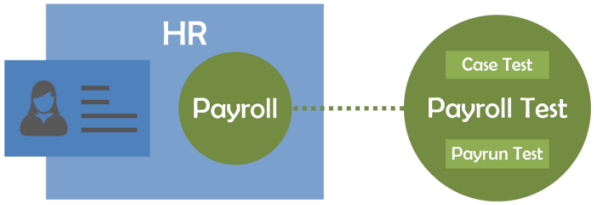
Test driven payroll software



The development of a scalable payroll application with automatic tests

In complex business applications such as payroll software, the development of new functionalities is a challenge. On the part of the users, requirements are formulated, and their fulfillment is verified. While requirements are formulated using requirements engineering, verification is difficult. Manual testing is very time-consuming and error prone. Existing test systems are often high-maintenance and computationally intensive and are decoupled from the requirements.

To enable automated testing of business cases, a uniform data and processing model that is decoupled from the application is required. The Payroll Engine provides such a scalable test system.

Using JSON or programming, the behavior of business cases and employee wage runs can be checked automatically. This also works for special cases such as retroactive mutations with back calculations or forecasts. For example, limit-tests can be used to check compliance with the minimum wage.

Tests are also an important tool in the development of Payroll Engine regulations to ensure the expected functionality of a product or a customer adaptation. Here, version-based regulation tests support release management.

Describing business cases through testing has enormous potential to significantly simplify and accelerate the development of business applications. This starts in the conception phase, where the test serves as a working specification for the implementation and verifies the release of the feature. During subsequent customizations, testing ensures that the feature continues to function correctly.

During operation, tests are used to describe a failure and verify that the failure has been corrected. By considering the error test in the future, a reoccurrence of the error can be detected early. Monitoring systems can use the Payroll API to analyze operations and generate test logs.

Setting up a test system is costly, but very rewarding. The result is a lean and agile development process and continuously increasing product quality. Requirements engineering benefits from documented and reusable tests.

Payroll becomes scalable in the sense that new functionalities can be developed at an unprecedented pace.