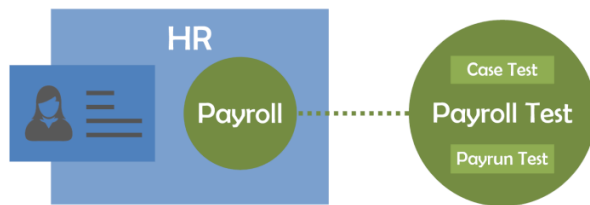


Test driven payroll software



The development of a scalable payroll application with automatic tests

For complex business applications such as payroll software, the development of new functions is a challenge. On the part of users, requirements are formulated and their fulfillment is verified. While requirements are formulated with requirements engineering, testing them is difficult. Manual testing is extremely time consuming and error prone. Existing test systems are often high-maintenance, computationally intensive, and disconnected from requirements.

To enable automated testing of business cases, a unified data and processing model, decoupled from the application, is required. The *Payroll Engine API* offers such an evolutionary test system.

Using JSON or programming, the behavior of business cases and employee pay runs can be tested automatically. This also works for special cases such as retroactive mutations with back calculation 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-related regulation tests support release management.

The description of business cases by means of tests 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 also verifies the release of the feature. In future adaptations, testing ensures that the feature continues to function correctly.

In operational mode, tests are used to describe a failure and verify that the failure has been corrected. Future consideration of the error test provides early detection of its recurrence. Tracking systems can use the Payroll API to analyze flows and generate test logs.

Building a test system is costly, but very rewarding. The result is a streamlined and agile development process, as well as continuously improving product quality. With continued use, requirements engineering benefits from documented and reusable tests.

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