Test Result Viewer

Graduate

Olivier Lischer

Luzia Kündig

Problem: Any kinds of tests in software are an important tool to improve software quality and robustness when changes must be made. Typically, automated tests can be run locally on the developer's machine or centrally using continuous integration, triggered by changes to a central repository. However, the benefits can only be leveraged if the test results are presented in a way that helps in the development process. In existing solutions that represent test results graphically it is often the assumption that test results are only relevant if something went wrong. In case of a successful execution, no details are presented to a developer except the final result "all passed". In the particular case at Sonova, running and evaluating tests is more complex than this. There are tests that depend on specific hardware to run, tests that run up to 100 times in one test execution, tests that only run during the night or on weekends because of their duration. The current test result visualization tools suffer from various deficiencies and need replacement.

Approach / Technology: Test results in different formats along with relevant metadata in a separate file are produced by the current infrastructure at Sonova. This data should be received, parsed and transformed into a unified schema using the Data Consumer component. Then the data is saved and displayed by the Test Result Viewer. According to our own research, some recommendations and experience present at Sonova, we chose Python for the Data Consumer and Elasticsearch and Angular to implement the Test Result Viewer.
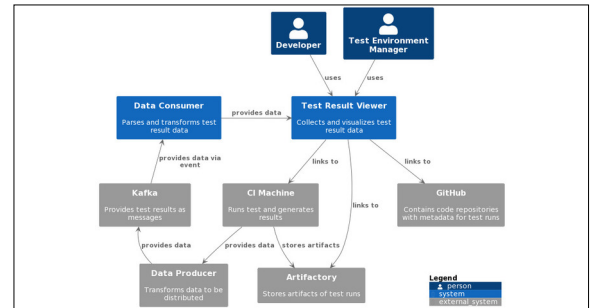
Conclusion: We implemented a platform to display test results in various views fulfilling Sonova's requirements. The common database schema we created largely depends on the data that is available

in the different result formats and the data that is needed by the users of this tool. By using a separate file to provide metadata it will be easy to add more information in the future. The visualizations we created require good knowledge of the Elasticsearch querying tools and response formats. The abstraction of these specifics inside the frontend allows creating additional views easily if needed.

Existing Tool by Sonova
Sonova AG

Architecture Context
Own presentment

Advisor
Thomas Corbat

Co-Examiner
**Guido Zgraggen,
Google Switzerland,
Zürich, ZH**

Subject Area
**Application Design,
Software**

Project Partner
**Sonova AG, 8712 Stäfa,
ZH**

Test Result Viewer
Own presentment