

# Towards Continuous Round-trip Engineering of Architectural Performance Models

## Motivation

Architectural performance models (APM) provide several benefits. On one hand, they represent a software on a higher level of abstraction than source code. This allows software developers to better comprehend a software. On the other hand, APMs can be used for a cost-efficient pro-active performance analysis by predicting performance characteristics. As a result, they can guide design decisions. In agile software development processes, models, such as APMs, are outdated fast due to the short development cycles and manual effort required to keep the models up-to-date.

Therefore, the Continuous Integration of architectural Performance Models (CIPM) approach was proposed to tackle these issues. In a model-based CI pipeline, changes from commits are analyzed to preserve the consistency between the source code and an APM. However, it does not consider changes in the APM yet. Software developers and architects can change the APM to, for example, adapt it for upcoming development cycles. These changes should be reflected in the source code as much as possible, while their manual changes should be not overwritten during the next commit change analysis.

## Task

In this Praxis der Forschung, the task is to research methods for the aforementioned issues: reflection of manual changes in the source code and preserving these manual changes. Then, suitable methods should be selected, prototypically implemented within the CIPM approach, and evaluated. Combined with the existing pipeline, this work contributes towards a continuous round-trip engineering of APMs.

If you are interested or have further questions, please contact:

**Martin Armbruster** (martin.armbruster@kit.edu)

Further information: <https://mcse.kastel.kit.edu/>