

# KIMAS: Kenngrößen- und Indikatoren-Managementsystem

Degree programme: BSc in Computer Science | Specialisation: Web and Business Applications

Thesis advisor: Prof. Dr. Olivier Biberstein

Expert: Dr. Eric Dubuis (Comet AG)

External project partner: Image Factory AG, Biel/Bienne

Key performance indicators (KPIs) assist enterprise executives in their decision-making process by providing quantified and comparable values derived from their business data. Well known KPIs are those coming from the financial community like net profit margin or sales growth. This project aims to improve an existing software system for management of KPIs which has been successfully running in productive environment for more than a decade.

1

## Project goals

The existing application, despite its many great features had two main flaws which were the annual change occurring at the fiscal-year end and its poorly implemented workflow pattern. The former entailed considerable manual maintenance effort and the latter made changes on indicator measurement- and assessment-periodicities very cumbersome. Instead of quick fixing existing code base, a new, completely rewritten application has been developed in the context of this bachelor thesis. Main goal of this project was to use modern development stack to make the resulting software as future-proof as possible.

## Software Archeology

Since the project was an improvement of a legacy system, the first phase was to understand the existing data model in order to reduce its complexity and improve overall design. Due to optimization of legacy application, its business logic resided partially in programming code and partially in database stored procedures. This made the understanding of data flow very time consuming. Eventually, a lean domain model has been developed to support all required business use cases while remaining as minimal as possible.

## Software Development Process

The resulting software was a product of applying as many agile software development practices as a one-developer team allows it. Requirements have been captured as user stories in dedicated JIRA project, which was also the go tool for sprint planning. Git branches were created for each user story and were in the end merged into the master branch. Continuous integration workflow as set up to compile and test application on each Git push.

## Development Stack

Since the main goal was to use future-proof technologies, the legacy monolith software was split into an Angular frontend and an ASP.NET Core backend powered by Entity Framework Core code first database approach. The latter enables to centralize business logic in the application code and delegate data access related tasks to the object relational mapper provided for that purpose. Communication between the REST API and the Angular powered frontend was achieved with token authenticated AJAX requests. This separation of frontend and backend makes the resulting software future-proof by making it possible to replace one of the two by a newer stack without breaking the other thus making software maintenance easier.



Eugene Kudryavtsev

