

# How to Enable a User-centric, Web-based Co-creation Process to Facilitate Software Demonstrator Development in Automotive Engineering

Lukas Greussing and Markus Zoier<sup>1</sup>

**Abstract.** Achieving optimal digital information supply from different automotive engineering domains has become an increasingly difficult task for supporting information management systems. Modern software engineering requires the collaboration of a variety of stakeholders – including the envisaged users – right from the beginning of a software project and rapid value delivery as a goal. Heterogeneous qualifications and insufficient software development skills escalate the requirements to the underlying framework and supporting platforms. One example to outline the challenge from above is the development of an engineering data management system for the automotive industry. In application-oriented research, experts from the automotive domain, development process, internal IT department and software suppliers usually have to design the envisaged IT solutions in an agile and interdisciplinary way. User-centric software demonstrator development can enable continuous concept exploration and allow ad-hoc user feedback for instant improvement, and is therefore a promising approach to cope with this major challenge.

From the software perspective, web technologies can help greatly to achieve this goal. But the overwhelming abundance of software framework choices and their usually strict separation of server and client development and technology base increase the complexity of projects and require in depth experience in frameworks. This fact often prevents end-users and semi-professionals from actively participating simultaneous in the software co-creation process. Promising

approaches to address this problem are full-stack isomorphic JavaScript frameworks. There are currently only a few up to date frameworks available in this domain, and one of them - Meteor<sup>2</sup> - has gained a lot of popularity within the web-development community. Meteor allows engineers to develop all application environments in one language, JavaScript. It embraces the JavaScript ecosystem and provides full stack reactivity, meaning the UI seamlessly reacts the true state of the application with minimal development effort.

A drawback of Meteor is the absent of a fully featured editor aiming end-user or semi-professionals demands and enabling effective rapid prototyping. This talk provides insights into a project at Virtual Vehicle, which was aimed at supporting a user-centric, web-based co-creation process to facilitate software demonstrator development at an automotive engineering company. It includes results from a respective master thesis, which was dedicated to building a web-based editor around Meteor to create web-based user demand driven demonstrators. The generated code, styles and application structure follow common Angular2 and Meteor recommendations. They can be easily customized and downloaded to provide an almost seamless transition between the early prototyping phase and sophisticated software pilots. Meteor in combination with the web-based editor can facilitate the co-creation process between end-users and professional software engineers. It also assists in the software deployment process by providing multiple strategies and supporting all major server architectures.

---

<sup>1</sup> Virtual Vehicle Research Center, Inffeldgasse 21A, 8010 Graz, email:{gruessing, zoier}@v2c2.at

---

<sup>2</sup> <https://www.meteor.com/>