How to Simplify Software Projects with Conceptual Model Driven Software Development (CMDSD)

Modern modeling techniques like MDA (Model Driven Architecture), MDSD (Model Driven Software Development), DSL (Domain Specific Languages) and others try to simplify the software development process and reduce project risks. Nonetheless, 70% of all software problems run into problems due to incomplete requirements, exaggerated expectations from users or too little actual involvement of domain experts during the specification phase. But must we be content with the state of affairs?

We present an extended version of MDSD called 'Conceptual Model Driven Software Development' (CMDSD) which focuses on the needs and skills of domain experts and allows to define immediately executable domain models. The use of the everyday knowledge of domain experts and the simple conceptual representation of underlying complex technologies are the foundations of CMDSD-based projects. In contrast to established technologies and methods, CMDSD concentrates on building rich domain models right away while specifying requirements - together with domain experts who must not have any IT expertise. Usually CMDSD allows full prototyping of moderately complex applications during a one-day-workshop together with domain experts. We will explain the underlying philosophy of CMDSD, show the theoretical foundations and present the methodologies required to build CMDSD-based applications. The ensuing reduction in project costs and risks and the raise in user acceptance of the resulting solutions prove the validity of the approach.

Keywords: Conceptual Model Driven Software Development, Domain Driven Design, Agile Modelling