Model-Driven Software Development (MDSD/MDD/MDA) («MDA»)

Often one tries to achieve a goal with models – also in software development. Get to know the possibilities to successfully design software development projects with different modeling techniques.

Duration: 1 day
Price: 850.–
Course documents: Digicomp course material.

Content

In this module, you will get to know the possibilities of model-driven software development. In the minds of the people involved in software development, there is a mental model of the software to be created. This is usually implemented by developers directly with the help of a programming language. Model-driven software development makes it possible to express this mental model at a higher, more abstract level in the form of formal models. By means of one or more transformations, such models are then converted into source files of one or more programming languages. Furthermore, it is also possible to interpret models at runtime instead of creating source files.

- Benefit promise of model-driven software development
- Subtypes of model-driven software development (model as documentation, code visualization, round-trip engineering)
- Definition of terms MDSD, MDD, MDA
- The step from software development to model-driven software development
- Modeling languages, Domain Specific Languages (DSL)
- What are meta-models and what are they needed for?
- Transformation, generation and interpretation techniques
- Teamwork in the context of model-driven software development
- Version control in the context of model-driven software development
- Requirements management in the context of model-driven software development
- Documentation in the context of model-driven software development
- Consideration of special software types (embedded, product lines)
- Overview Modeling and Generation Tools
- What to look out for in model-driven software development
- Procedure: Introduction, Terms, Modeling Part I, Meta-Modeling, Syntax
- UML, Modeling Part II, Model Evaluation, Software Architecture, Tools
- Development Process, Team Roles

Key Learnings

- Assessing the benefits and limitations of model-driven software development
- Familiarity with the different modeling and generation options
- Different types of Domain Specific Languages (DSL) and their suitability
- Meaning and differences of model, meta-model, and meta-meta-model
- Overview of a selection of the most important modeling and generation tools

Target audience

This course is aimed at IT developers and IT architects who want to be able to assess the individual advantages of model-driven software development themselves and want to gain an overview before using model-driven software development.

Requirements

Please bring your own laptop.

No knowledge of programming languages is required. Initial experience with modeling in general, e.g. with the Unified Modeling Language (UML), is useful. Comprehensive knowledge of SW architecture in the scope of the following course is required:

- iSAQB® Certified Professional for Software Architecture – Foundation («CPSAFL»)

Further courses

- Sparx Enterprise Architect – An easy entry-level («SEAELG»)
- UML Professional 2 Foundation («UMLZFP»)
- Unified Modeling Language («UML»)
- Service-oriented architecture («SOA»)

Any questions?

We are happy to advise you on +41 44 447 21 21 or info@digicomp.ch. You can find detailed information about dates on www.digicomp.ch/courses-software-engineering/software-engineering-modelling/course-model-driven-software-development-mdsdmddmda