



## Component Manager

With the Component Manager machines and systems can be structured modularly, so that customer-specific differences (other software module combinations, additional modules, etc.) can be created without modification and thus without test of the existing software. The component orientation thus enables creation of the overall function of the automation system at the time of instantiation by interconnecting autonomous software modules.

Real machines or systems consist of a variety of subsystems that can be structured according to different criteria (e. g. affiliation with a trade, function). Through the structuring in subsystems it is possible to disregard the complexity of the overall system and only deal with the (reduced) complexity of the subsystem and its interface.

The M1 controller makes the concept of software modules available for software-side organization of these subsystems. Software modules communicate via a uniform interface and can be created in every programming language available on the M1 controller.

A component abstracts its internal structure or its implementation details and describes itself completely through the defined interfaces and properties.

Components are individually compiled, versioned, tested, and if necessary documented. They can be interconnected with each other and with the I/O system to create the overall function of an

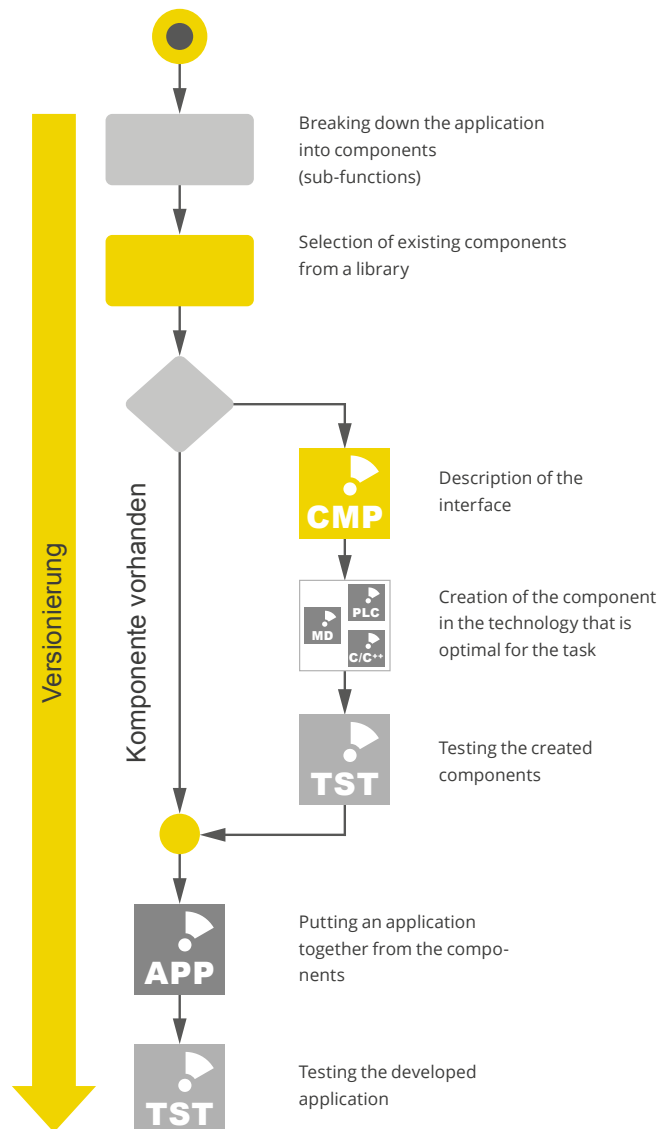
automation system. Implementation of the functionality and the application of the functionality are decoupled.

This procedure enables a multi-level work process:

- Describe a target programming language (interface description) independently of components
- Program components on the basis of the interface description
- Develop applications by interconnecting components (and I/O system if necessary)

Component Manager makes the appropriate tools available for all phases.

- Features:
- The technical process is in the foreground
- Complete abstraction of the internal structure of the components
- Separation of definition and implementation
- Convenient description language for the interface
- Code generation for low-level areas
- Graphic application development
- Clear and understandable application presentation in the SolutionCenter
- Support of functionality on the application level



## Component Manager modules



**Graphic application development**  
Interconnect graphically



Component **test**



**Component Designer**  
Technology-independent description of the component interface



**Developer** for C/C++, PLC, MATLAB® / Simulink®