QA Concepts

Testing supplies information about the quality of the service and whether if fulfills the requirements defined during the conceptional phase of the development - with the goal of activating the service with as few bugs as possible.

We support testing by the following concepts and tools:

Runtime Analysis and Troubleshooting of Distributed Systems	With the **Trace Analyzer** as a part of the **Scheer PAS** *ANALYZER* it is possible to run test cases and to trace through the UML process flow afterwards. This applies to synchronous processes as well as to asynchronous processes. For more information on tracing services refer to the Analyzer User's Guide.
Model Debugging	The **xUML Service Interactive Debugger** is part of the **Scheer PAS** *BUILDER* and its **SOAP Test Tool**. The Interactive Debugger integrates seamlessly into MagicDraw and allows to trace visually through a UML model, manage breakpoints and watch objects. For more information on the Interactive Debugger refer to the Builder User's Guide
Regression Tests	The **Regression Test Tool** as part of the **Scheer PAS** *ANALYZER* provides the possibility to build up full test scenarios. By that approach reproducible tests lead to a defined service quality after the service having been changed. For more information on regression testing refer to the Analyzer User's Guide.
Backend Mockups	As with testing not all backends are always available reliably, it is possible to record or manually build up backend access data to simulate backends for the use with test cases. For more information on backend mockups refer to the Builder User's Guide .
Unit Testing	The **Scheer PAS** *BUILDER* offers the possibility perform unit tests. Single classes can be stereotyped as to be testable, so that the class implementation can be tested independently from the rest of the service implementation. For more information on the concept of testable classes refer to the following sections.
Continuous Delivery	The **Scheer PAS** *BRIDGE* offers a list of command line tools to help you integrate your UML model development into a continuous delivery scenario, such as e.g. Jenkins. For more information, refer to Continuous Delivery with the Bridge.

In the next chapters, you will find some general explanations on testing of services running on the Bridge and more details on the concept of making classes testable.