MES QUALITY COMMANDER User Guide

Release 4.2

Model Engineering Solutions GmbH

Apr 22, 2020

TABLE OF CONTENTS

1 What is MQC and how can it help you? 1

1.1 What is MQC? . . . 1

1.2 From data to quality . . . 2

1.3 How MQC supports quality assurance and quality improvement . . . 4

2 Highlights in MES Quality Commander® (MQC) 7

2.1 MES Quality Commander® (MQC) v.4.2 . . . 7

2.2 Highlights History . . . 7

2.2.1 MES Quality Commander® (MQC) v.4.1 . . . 7

2.2.2 MES Quality Commander® (MQC) v.4.0 . . . 8

2.2.3 MES Quality Commander® (MQC) v.3.5 . . . 8

2.2.4 MES Quality Commander® (MQC) v.3.4 . . . 8

3 Installation 11

3.1 Install MQC Editor . . . 11

3.2 Change Default Password . . . 11

3.3 System Requirements MQC Editor . . . 12

4 Quick Start Guide 19

4.1 Start MQC . . . 19

4.2 Create new Analysis . . . 20

4.3 Check Data Availability . . . 21

4.4 Import Project Structure . . . 22

4.5 Check Data Status . . . 23

4.6 Import Quality Model . . . 23

4.7 Check Quality Trend . . . 24

4.8 Check Quality Status . . . 26

4.9 Work with MQC . . . 26

5 Dimensions and Structures 31

5.1 Artifacts and Artifact Structure . . . 31

5.2 Revisions . . . 32

5.3 Projects and Milestones . . . 32

5.4 Measures and Measurements . . . 34

5.5 Derived Measures . . . 37

6 Quality Concept 39

6.1 Quality Computation . . . 39
6.2 Quality Bins . . . 41
6.3 Quality Model Structure . . . 42
6.4 Quality Aggregation . . . 42

7 Data Sources **45**
7.1 MES Model Examiner® (MXAM) . . . 45
7.1.1 MXAM Data imported by MQC . . . 45
7.2 MES M-XRAY® (MXRAY) . . . 46
7.2.1 MXRAY Data imported by MQC . . . 46
7.3 MES Test Manager® (MTest) . . . 48
7.3.1 MTest Data imported by MQC . . . 48
7.4 Manual data import . . . 49
7.5 PikeTec TPT . . . 49
7.6 Razorcat Tessy . . . 50
7.7 MathWorks Polyspace . . . 51
7.7.1 Polyspace Xml-adapter . . . 51
7.7.2 Polyspace Text-adapter . . . 52
7.8 BTC EmbeddedTester . . . 53
7.8.1 Standard XML report format . . . 53
7.8.2 MQC XML report format . . . 55

8 Actions and Configuration Menu **57**
8.1 Actions . . . 57
8.1.1 Refresh Data . . . 57
8.1.2 Create Report . . . 57
8.1.3 Data of the Quality / Filtered by Quality . . . 58
8.2 Configuration . . . 58
8.2.1 Data Sources . . . 58
8.2.2 Project structures . . . 58
8.2.3 Quality Model . . . 58
8.2.4 Target Values . . . 58
8.2.5 Settings . . . 62
8.2.6 Pages . . . 63

9 Pages and Visualizations in Detail **67**
9.1 The MQC dashboard . . . 67
9.2 Data pages . . . 68
9.2.1 Data Trend . . . 68
9.2.2 Data Status . . . 68
9.3 Quality pages . . . 69
9.3.1 Quality Status . . . 69
9.3.2 Quality Sunburst . . . 69
9.3.3 Quality Heatmap . . . 70
9.3.4 Quality Trend . . . 70
9.4 Tool pages . . . 70

9.4.1 MXAM Tool Page . . . 70
9.4.2 MXRAY Tool Page . . . 72
9.4.3 MTest Tool Page . . . 76
9.4.4 Polyspace Tool Page . . . 78
9.5 Color Schemes . . . 82
9.6 Marking . . . 83

10 Project Structures Configuration **87**
10.1 Project Naming . . . 87
10.2 Artifact Mapping . . . 88
10.3 Project Milestone Structure . . . 88
10.4 Artifact Structure . . . 90
10.5 Artifact Weights . . . 91

11 Quality Model Configuration **95**
11.1 Base Measures and Default Values . . . 95
11.2 Derived Measures . . . 97
11.3 Quality Properties . . . 98
11.4 Quality Property Weights . . . 101
11.5 Context Categories . . . 102
11.6 Quality Model Kit . . . 103
11.7 Quality Model for MES data sources . . . 103
11.7.1 Quality Model for MXAM . . . 103
11.7.2 Quality Model for M-XRAY . . . 104
11.7.3 Quality Model for MTest . . . 104
11.8 Measurement function definition . . . 105

12 Validation of Configuration **107**
12.1 Validation of Artifacts . . . 107
12.2 Validation of Measures . . . 107
12.3 Validation of Context Categories . . . 107

13 Data Propagation **109**
13.1 Motivation . . . 109
13.2 How to Propagate Data? . . . 109
13.3 Propagate Data For The Entire Project . . . 110
13.4 Propagate Data Towards Next Milestone . . . 110
13.5 Remove Propagated Data . . . 112

14 Context Categories **115**
14.1 Define your Context Categories . . . 115
14.2 Add whitelists/ blacklists to your Context Categories . . . 116

15 Target Values **121**
15.1 Configuration of Target Values . . . 121
15.2 Calculation of Target Values per Revision . . . 124

16 Custom Pages **127**
16.1 Custom Page Structure . . . 127
16.2 Custom Page Configuration . . . 129
16.3 Configuration Data Import and Page Creation . . . 131

17 Data Details **133**
17.1 Data Details Drill Down . . . 133
17.1.1 Data Selection for Drill Down . . . 134
17.1.2 Drill down to Data Trend . . . 134
17.1.3 Drill down to Data Status . . . 134
17.2 Tool Pages . . . 137
17.2.1 MXAM Details Page . . . 137
17.2.2 MTest Details Page . . . 139
17.2.3 M-XRAY Details Page . . . 139
17.3 Custom Pages . . . 139

18 Status Report **141**
18.1 Configuration of the Status Report . . . 141
18.1.1 Artifact selection . . . 141
18.1.2 Revision selection . . . 143
18.1.3 Report Type . . . 143
18.1.4 Output Format . . . 144
18.2 Content of the report . . . 144
18.2.1 Report and Project Information . . . 144
18.2.2 Quality Overview . . . 144
18.2.3 Data Availability Overview . . . 144
18.2.4 Data Trend Overview . . . 144
18.2.5 Data Status Overview . . . 144
18.2.6 Artifact Details . . . 145
18.2.7 Quality Property Details . . . 145
18.2.8 User Configured Custom Pages . . . 145
18.2.9 Configuration Information . . . 145

19 Analysis Setup Configuration **147**
19.1 How to use . . . 147
19.2 Configuration . . . 148

1 WHAT IS MQC AND HOW CAN IT HELP YOU?

1.1 WHAT IS MQC?

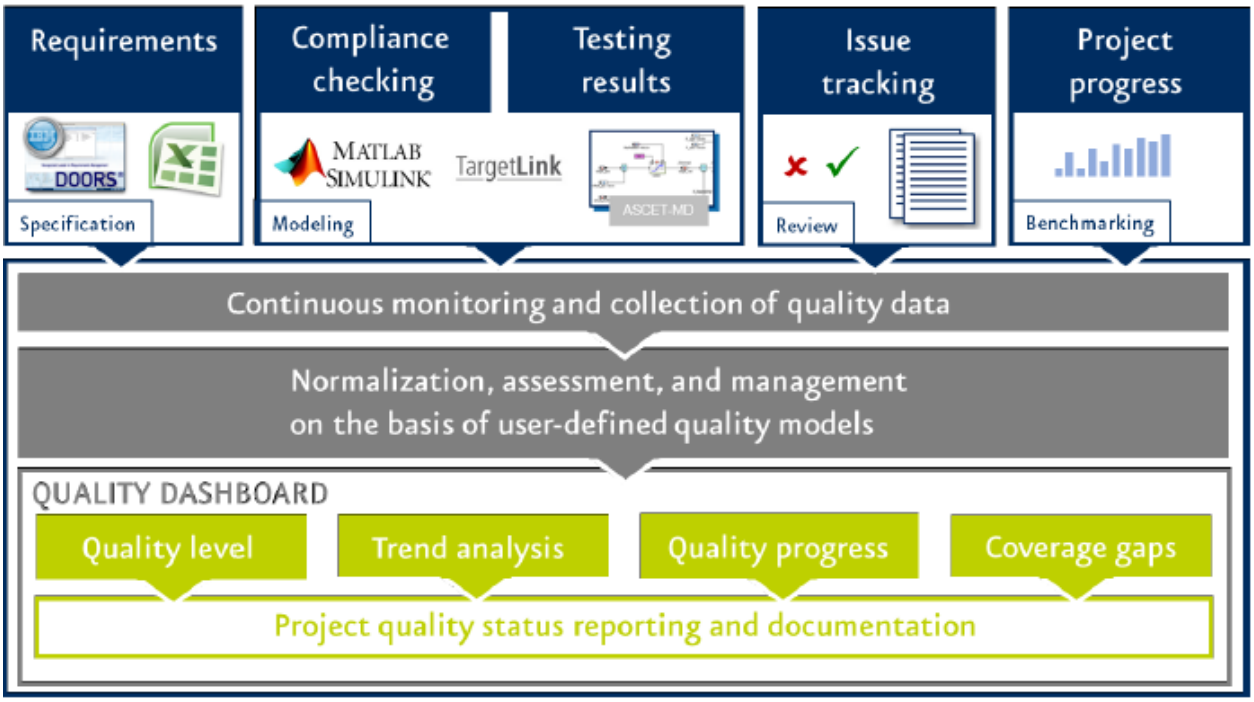


Figure 1.1: Quality assurance and monitoring with MQC

MES Quality Commander® (MQC) is a dynamic quality monitoring and management tool for software development that captures all the decision-making data that you need throughout the software life cycle. MQC computes and evaluates the quality and product viability of your software, based on relevant development artifacts and the corresponding key performance indicators (e.g. guidelines, complexity, tests, coverage and reviews). User-friendly visualizations of product maturity, weaknesses and need for action during the different stages of a project increase the software's development and product value.

MQC also optimizes return on investment by perpetual availability of trend analysis that indicates the product's achievable level of quality. An efficient visualization of quality and progress for different development projects ensures error proofing very early. Project-specific evaluation with individually configurable quality models adaptable to ISO 26262 or ASPICE enables quality assurance of safety relevant software development.

MQC provides different possibilities of reporting such as the editor itself or the web viewer for sharing information. Thus, effort and charges can be controlled and minimized. The web viewer guarantees multiple users to be able to access your project and supplies interactive reporting along with other features. Therefore, data discovery and operational reporting yield an entire understanding of the data's quality impact. MQC data import supports several operational tools and export formats, which allows a fast and easy setup of quality monitoring projects. Data collection can also be automated and integrated into continuous integration to make full use of your existing infrastructure and workflows.

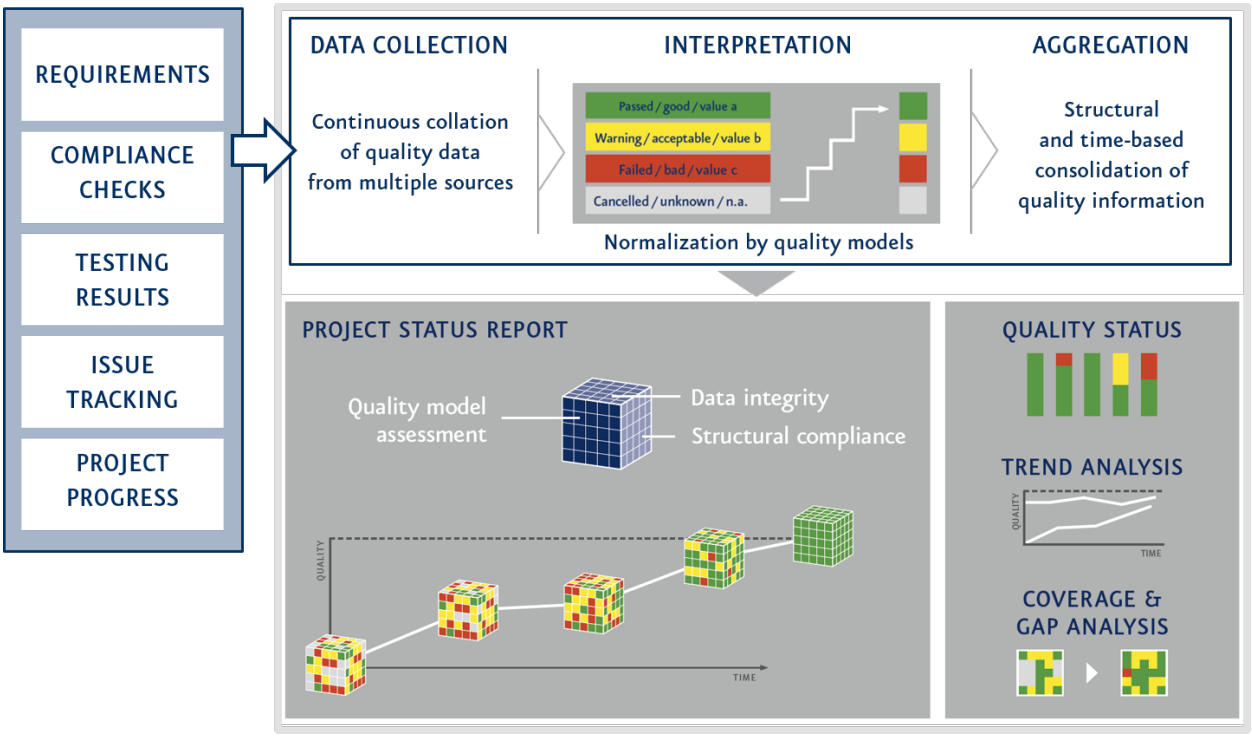


Figure 1.2: Data to quality: MQC's representation of the Software life cycle

Referring to the V-Model, MQC aims to be the Master tool of monitoring and managing your (model-based) Software Development Lifecycle (SDLC). As shown in Figure 1.3, MQC can collect data from various quality assurance activities at each stage of the SDLC. MQC not only provides efficient extraction of data from MES Tools, but also from other tools like TPT, Tessy, Polypace and Embedded Tester.

1.2 FROM DATA TO QUALITY

The project creation workflow can be separated into two parts.

The first and most important part to create an MQC project is "Data and project structure". It contains two processes: "Analysis" and "Creation". During the "Analysis" process, one has to analyze the available data with respect to the data's relevance of quality. In particular, the "7Ws" of data analysis (who, where, when, what, why, how and how many) are fruitful to answer to understand the data's structure. For more details, we refer the reader to the book "Agile Data Warehouse Design" (Lawrence Corr with Jim Stagnitto, 2014, pp. 31f).

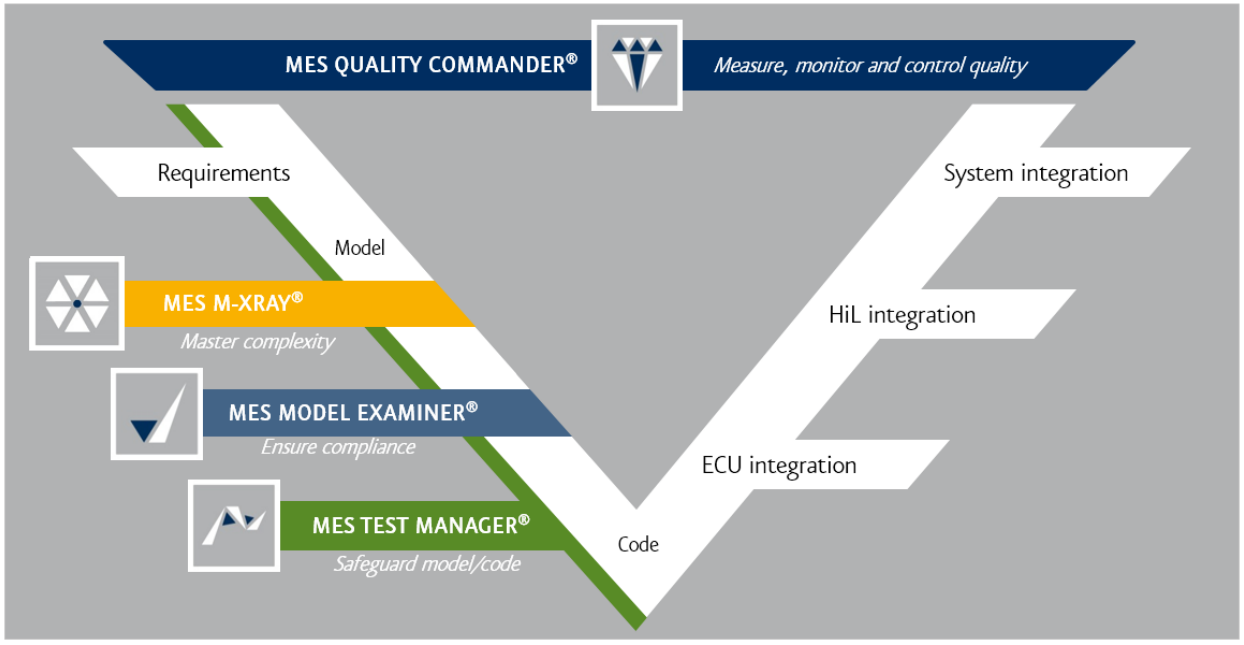


Figure 1.3: MQC evaluates data from a wide range of report-generating tools.

After having answered these questions, you should have identified relevant data. The next crucial step is the identification of MQC structures in your data. Such structures can be artifacts and hierarchies, data sources and data values, projects and milestones and much more.

After identifying the structures in your data, they need to be mapped onto the MQC structures. Particularly, this is a very abstract step, because it is a priori vague what "mapping" means in detail. Typically, MES quality and data experts, together with customer process experts, analyze the customer process and provide professional support to deduce MQC structures. All these steps form the foundation of the second process – "Creation".

First note that the order of the sub-processes of "Creation" as depicted in Figure 1.4 may be adapted, but we recommend the given sequence. Since "Artifacts" are those objects for which data arises and for which MQC does quality computation, it is a good idea to start with structuring these objects first. This structuring is an outcome of the "Analysis" process. Secondly, you should define the data source structure, which follows directly from the first process, too. As data sources consider the objects that yield data for the Artifacts, it is natural to create them after creating the Artifact structure.

The next step is creating context categories. By a context category, one is able to connect Artifacts and Data Sources. Vividly spoken, a context category provides information about which data is expected for the Artifacts. Additionally, context categories represent an MQC dimension for which filtering is possible so that the user can have multiple views on his data.

The definition of the project's time structure (so-called Revisions) is crucial to provide a chronological sequence to the changes of data.

The second part of the project creation workflow, "Use", is the application of the Data and project structure creation. It consists of (possibly automated) recurring data import, which is the basis for visualizations in MQC. After having imported data, you can work with the visualizations and perform data discovery. If the

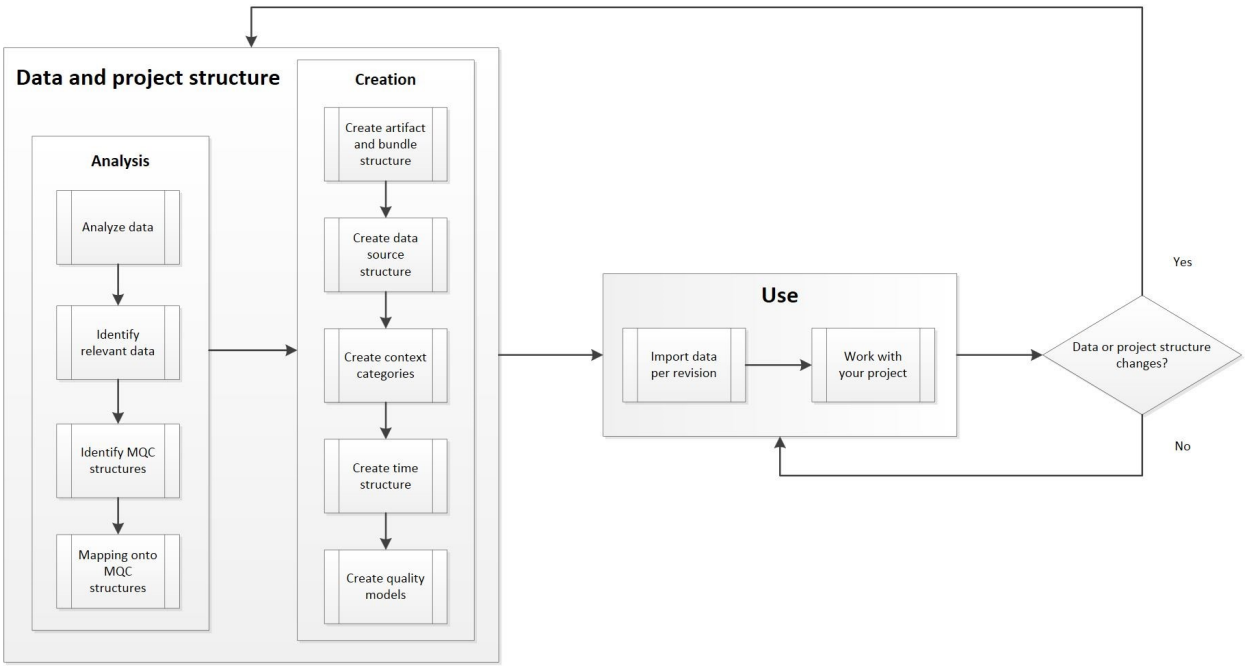


Figure 1.4: MQC Project creation workflow

data or project structure changes, you have to adapt them before importing new data. This entire part deals with quality computation and is a direct result of the previous part. The visualizations appear automatically.

1.3 HOW MQC SUPPORTS QUALITY ASSURANCE AND QUALITY IMPROVEMENT

The following illustration provides a coherence of high-level requirements, quality computation and aggregation, as well as (product) quality.

The grey box in the middle of Figure 1.5 symbolizes that the main purpose of MQC is to increase the (software) product quality.

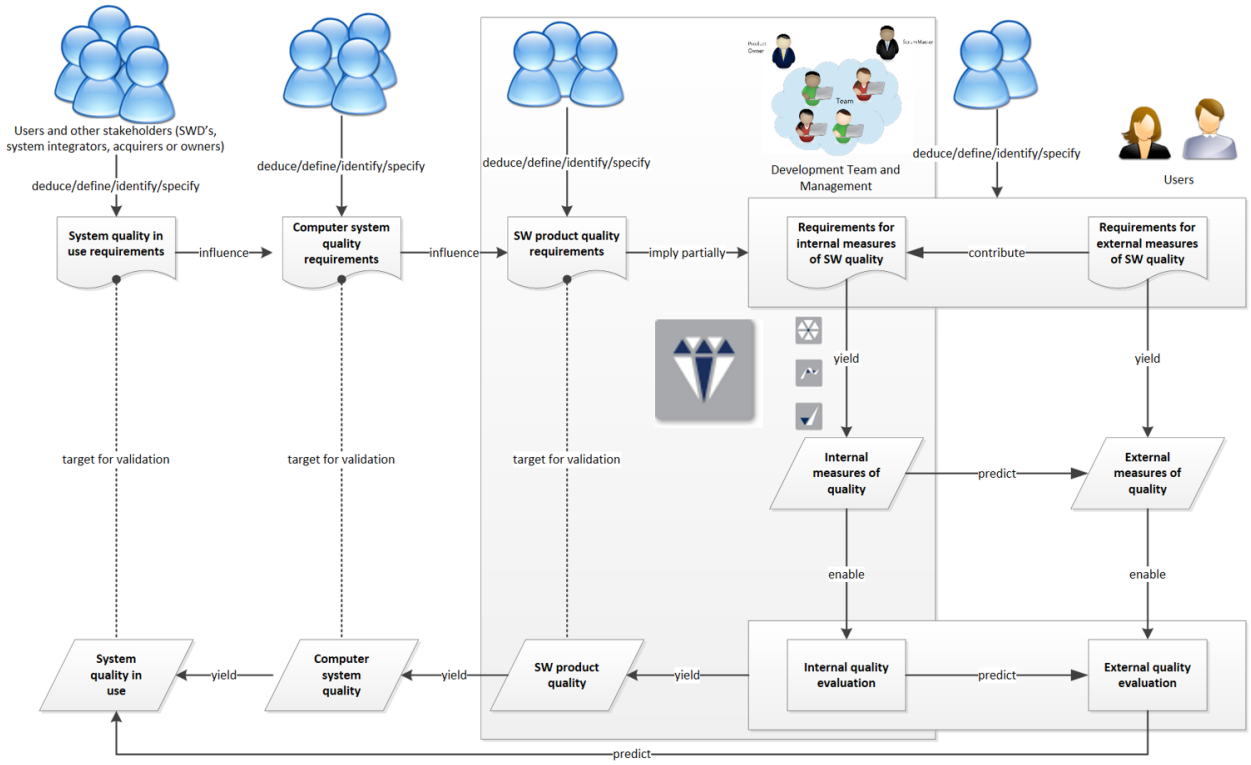


Figure 1.5: MQC Quality Life Cycle Model

2 HIGHLIGHTS IN MES QUALITY COMMANDER® (MQC)

2.1 MES QUALITY COMMANDER® (MQC) V.4.2

Significant Import Time and Memory Reduction for Huge Data

The import of huge sets of data files and complete directories of files is much faster now. In addition, the memory consumption was reduced considerably.

The mechanism of monitoring for file and directory changes was reworked to detect changes safe and with less computing resources. Due to the adapted caching mechanism the reimport of changed files is much faster now as well.

User Experience

The marking in the bin distribution is more intuitive now. Revisions on status pages can be selected with just one click. Similarly, a quality bin on trend pages can be selected for all revisions with one click.

The extension of marking when combining marking on bin distribution and in KPIs is more intuitive now.

The showing of the KPIs and the calculations inside the KPIs are fully dynamic now and reflect the marking on the other visualization of the page.

Artifact and revision marking on quality or data pages is directly reflected on other quality and data pages.

A full description of these mechanismns can be found in *Marking*.

Availability of MQC Server and MQC Web Viewer/Editor

The MQC Server is now available for on-premise installation and setup. Now you can have a secure library of your analysis including access to your local network (and data).

With this availability of the internal library you can easily create a shared analysis (and some tasks can run regularly on the server, i.e. to update an analysis with new data)

Together with the MQC Server, the MQC Web Viewer/Editor is now available. It provides the functionality of the client directly in a browser while the analysis is opened/running on the MQC Server.

2.2 HIGHLIGHTS HISTORY

2.2.1 MES Quality Commander® (MQC) v.4.1

Improved Configuration Dialogs

MQC settings dialogs have been redesigned for better usability, and are available in MQC editor as well as MQC web viewer.

For more information refer to chapter *Actions and Configuration Menu*.

Fine grained configuration of Context Categories

Context categories define the relations between artifacts and data sources, measurements, and base measures. They help to define and focus on the artifact-relevant data from the available data pool.

For more information refer to chapter *Context Categories*.

2.2.2 MES Quality Commander® (MQC) v.4.0

Report generation project status

The new status report in html format documents all details of an MQC project, including graphics and context-specific tables to structure the information. The structure and extent of the report is configurable.

For more information refer to chapter *Status Report*.

Configure expected data via context categories

Improved tool adapter and tool page for Polyspace (The Mathworks)

For more information refer to section *MathWorks Polyspace*.

Extended tool adapter for TPT (Piketec)

For more information refer to section *PikeTec TPT*.

2.2.3 MES Quality Commander® (MQC) v.3.5

From quality to data details

Via the context menu, selected quality properties or artifacts link to detailed data specific to the selection.

For more information refer to section *Data Details Drill Down*.

Introducing artifact weights for quality aggregation

The artifact weights, as well as the weights for quality measures can be visualized in a hierarchical heat map, where the impact of an artifact and a quality measure scales with the size of a colored tile.

For more information refer to section *Artifact Weights*.

Additional tool pages for MES M-XRAY and MES Test Manager

For more information refer to section *MES Test Manager® (MTest)* and to section *MTest Tool Page*.

2.2.4 MES Quality Commander® (MQC) v.3.4

Target value configuration for measures and quality properties

For each measure or quality property one or multiple target values (thresholds, expectations) can be defined per project milestone.

For more information refer to chapter *Target Values*.

Introducing quality property weights for quality aggregation

For each quality property a weight can be defined to be used in the aggregation of quality properties to sub-characteristics in the structure of the quality model.

For more information refer to chapter *Quality Property Weights*.

Tool adapter for Polyspace (The Mathworks)

3 INSTALLATION

In this chapter, we will guide you through the installation procedure for the MQC editor.

3.1 INSTALL MQC EDITOR

To install MQC, you must download the required Windows installer. You received or will receive an email from MES containing a link and the necessary credentials to download the installer file from our server.

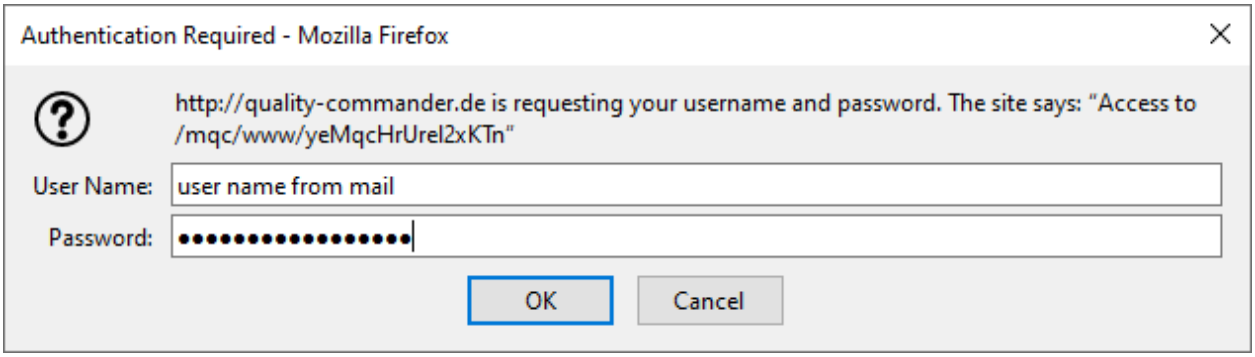


Figure 3.1: Type in the credentials and press OK

Please note that you must be able to unzip the files.

After extracting the zip-file, navigate to the subfolder: `\TIB_sfire-analyst_10.3.3_single_user_win\Products\TIBCO Spotfire Installer`.

Now open the file `setup-single-user-10.3.3.exe` and proceed with the next step.

When installing MQC, the login to the MQC server is mandatory. This step will validate your MQC license and will provide you with the latest updates of the current version of MQC.

Please note that you must necessarily log in at least once every 30 days. This makes sure that your MQC license is revalidated and you can take advantage of updates, bug fixes and further improvements.

3.2 CHANGE DEFAULT PASSWORD

If you would like to change the default password provided by MQC, please open a browser and navigate to http://mes-qualitycommander.com. Login with your personal credentials.

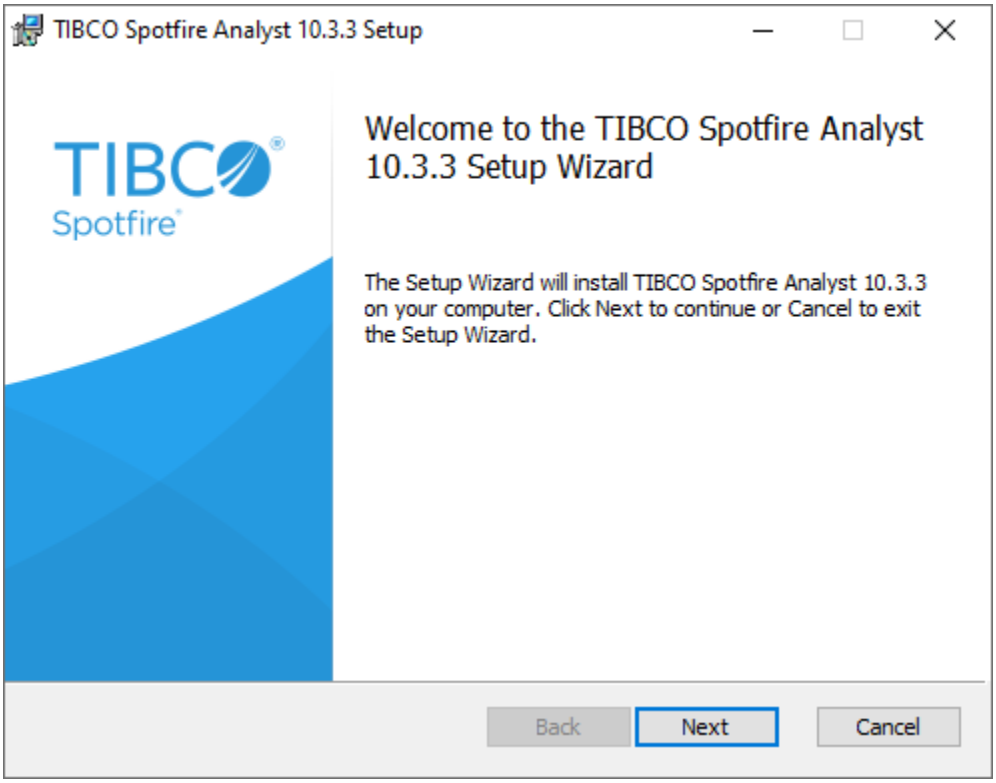


Figure 3.2: In the installation wizard, press Next.

Then inside your browser click on your user name at the top-right of the window and choose 'My account'. This opens the Administration Console for your account. Select 'Change password' and follow the instructions of the 'Change TIBCO Spotfire password' dialog.

3.3 SYSTEM REQUIREMENTS MQC EDITOR

Note: MQC Editor installation includes TIBCO Spotfire Analyst installation

Table 3.1: System requirements for the MQC editor Hardware

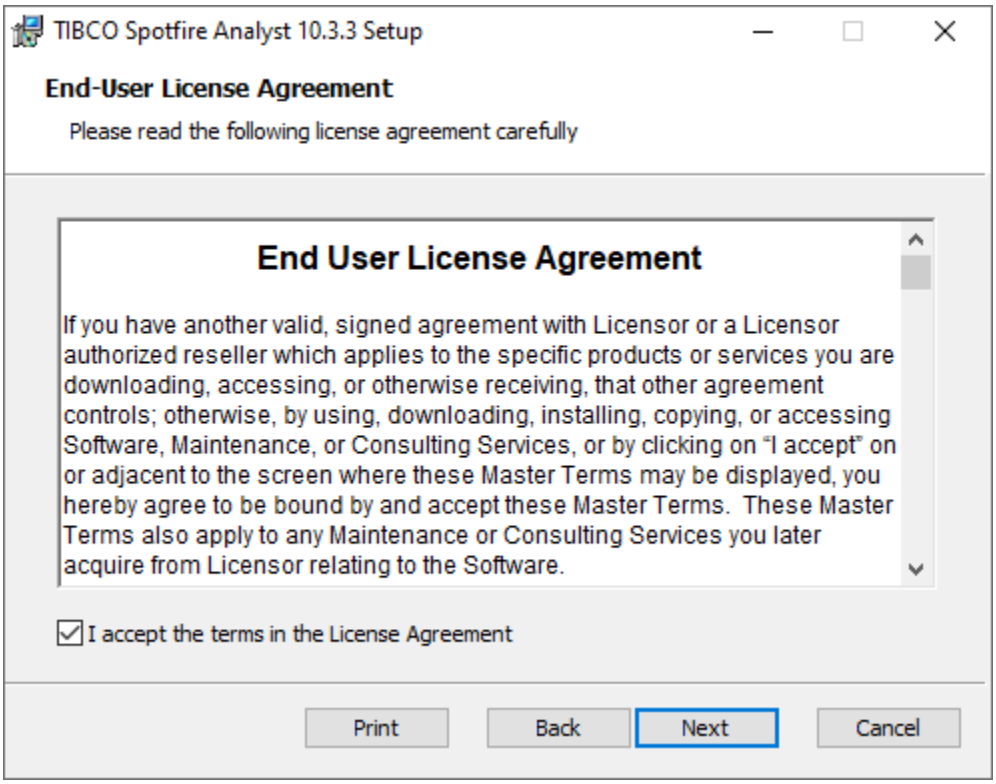


Figure 3.3: Please read the license agreement carefully, accept it using the check box and click on Next.

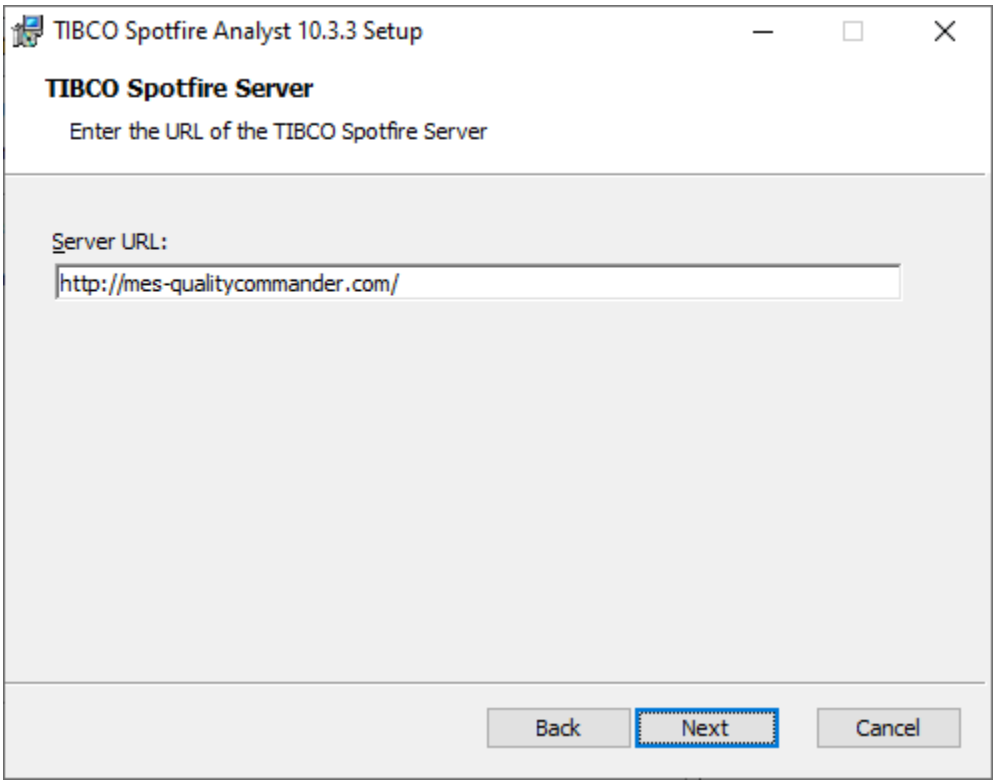


Figure 3.4: Enter the URL http://mes-qualitycommander.com/ as server address and press Next.

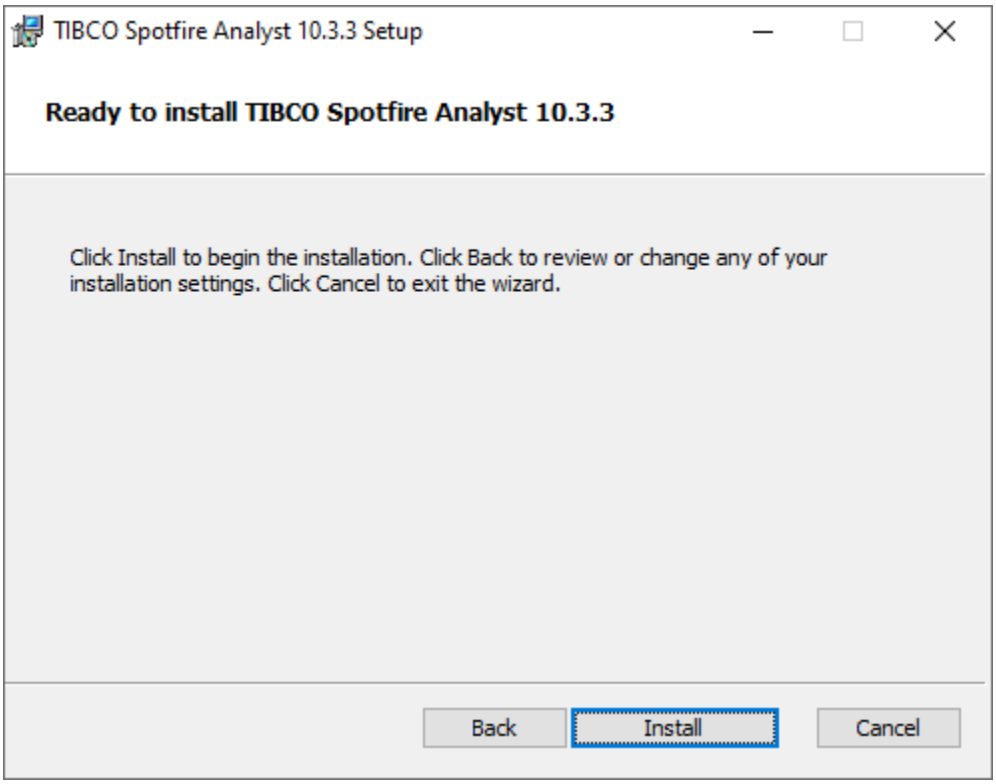


Figure 3.5: Finally to install the program, press Install.

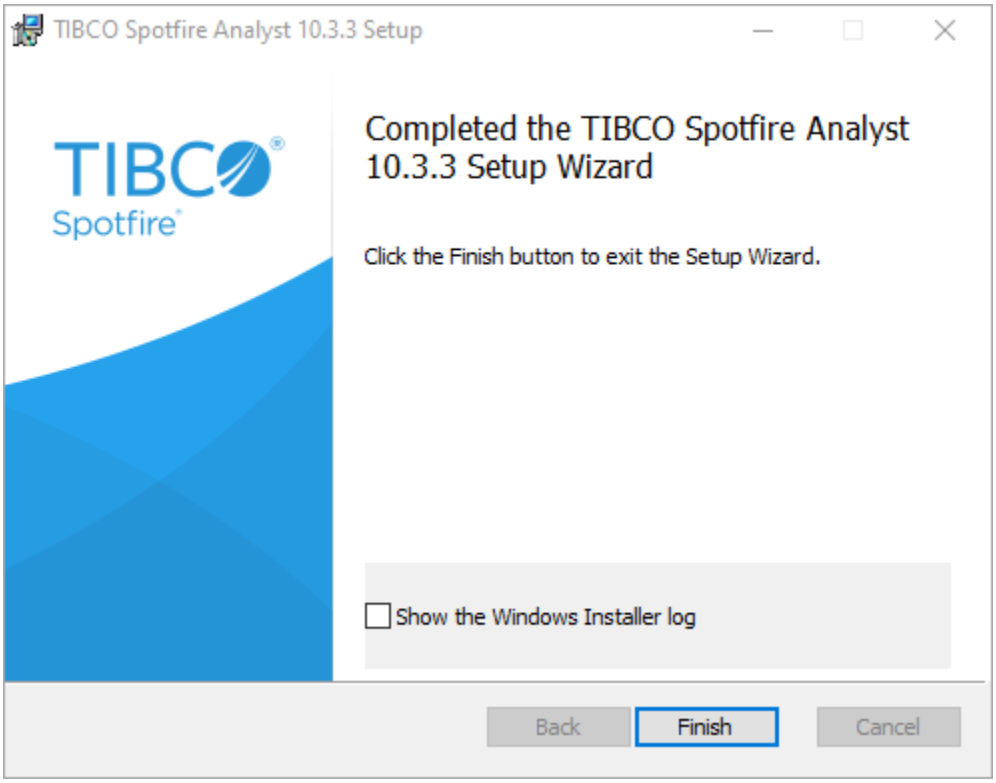


Figure 3.6: The installation may take a few minutes. Afterwards, please confirm with Finish.

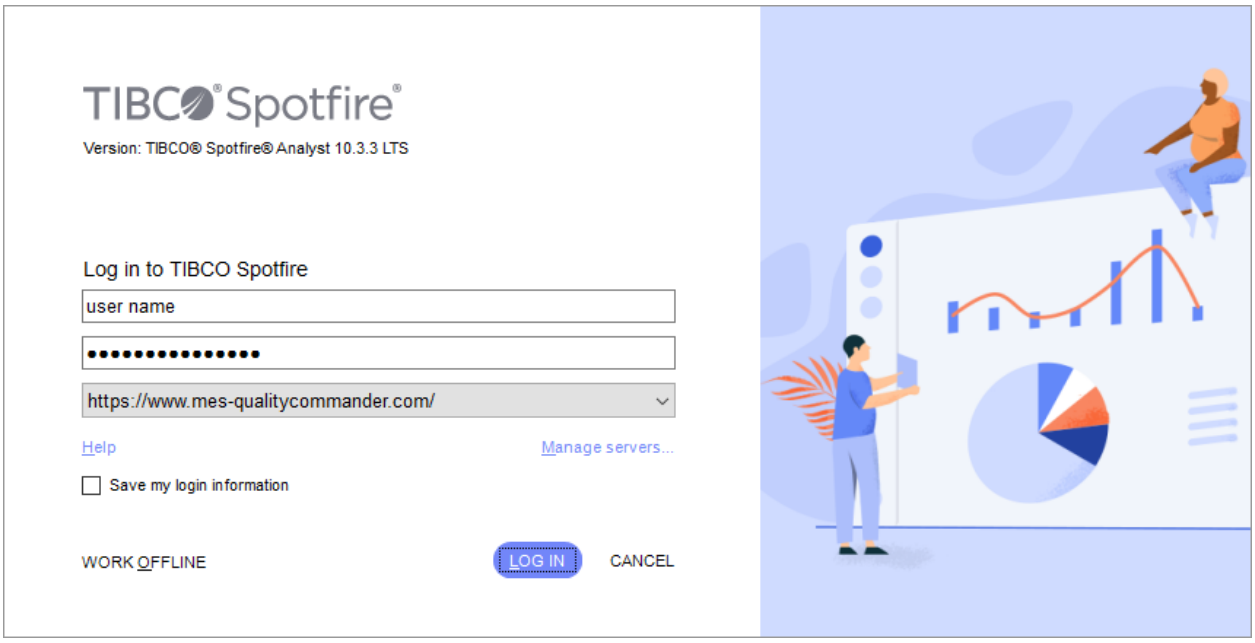


Figure 3.7: Please, open Spotfire and login with your personal credentials using the username and password provided to you by MES. Please note the lowercase and press Log In.

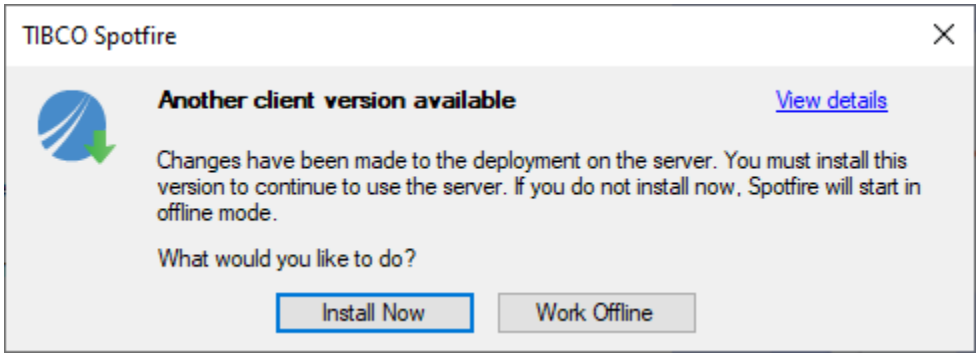


Figure 3.8: Please confirm the requested update by pressing Install Now

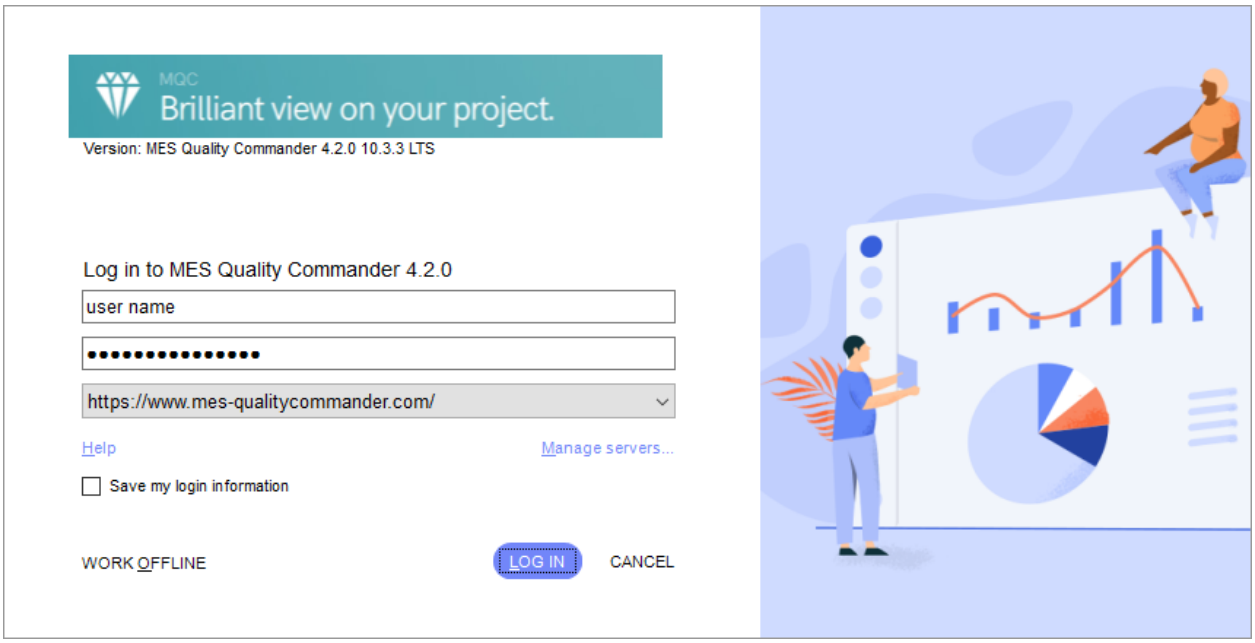


Figure 3.9: Once the update is complete, you can now start MQC by logging in with your credentials as mentioned above. Click on Log In

Category	Requirement
Hardware	
Processor	Minimum: 2 Cores, 2 GHz Recommended: 4 Cores or more (Intel Core i5 or equivalent), 2+ GHz, 64-bit
RAM	Minimum: 4 GB Recommended: 8 GB or more Note: Large data sets can require more RAM
Hard disk space	10 GB is recommended for installation and normal use.
Display	Minimum: 1024x768 pixel resolution, 16 or 32-bit color depth. Recommended: 1920*1080 pixel resolution or higher, 32-bit color depth.

Table 3.2: System requirements for the MQC editor Software

Category	Requirement
Software	
Operating System	Microsoft Windows 10 Microsoft Windows 8, 8.1 Microsoft Windows 7 64-bit
Installation Permissions	Administrator rights are required.
Microsoft .NET Framework	Microsoft .NET Framework version 4.5.x, 4.6.x, 4.7.x. NOTE: Microsoft .NET Framework 4.6 (or higher) is strongly recommended. NOTE: If Microsoft .NET Framework 4.5 isn't installed when running the installer, the installer can (if the users accepts this when prompted) download and install it.
Microsoft Office (Optional)	Microsoft Office 365 Microsoft Office 2016, 32- and 64-bit versions Microsoft Office 2013, 32- and 64-bit versions Microsoft Office 2010, 32- and 64-bit versions NOTE: Microsoft Office needs to be installed in order to use functionality that integrates with Office products (such as importing data from Excel or Access, or exporting to Power Point)

4 QUICK START GUIDE

This chapter will give you a quick overview of the functionality of MQC and will show you how to use it to analyze your project quality and data.

Note: The Quick Start Guide is based on the sample data within a zip file called `MQC_QuickStart.zip`. You can download the sample data here.

Please unzip this file to an appropriate directory for your sample data.

4.1 START MQC

Start MQC by clicking on the TIBCO Spotfire icon.

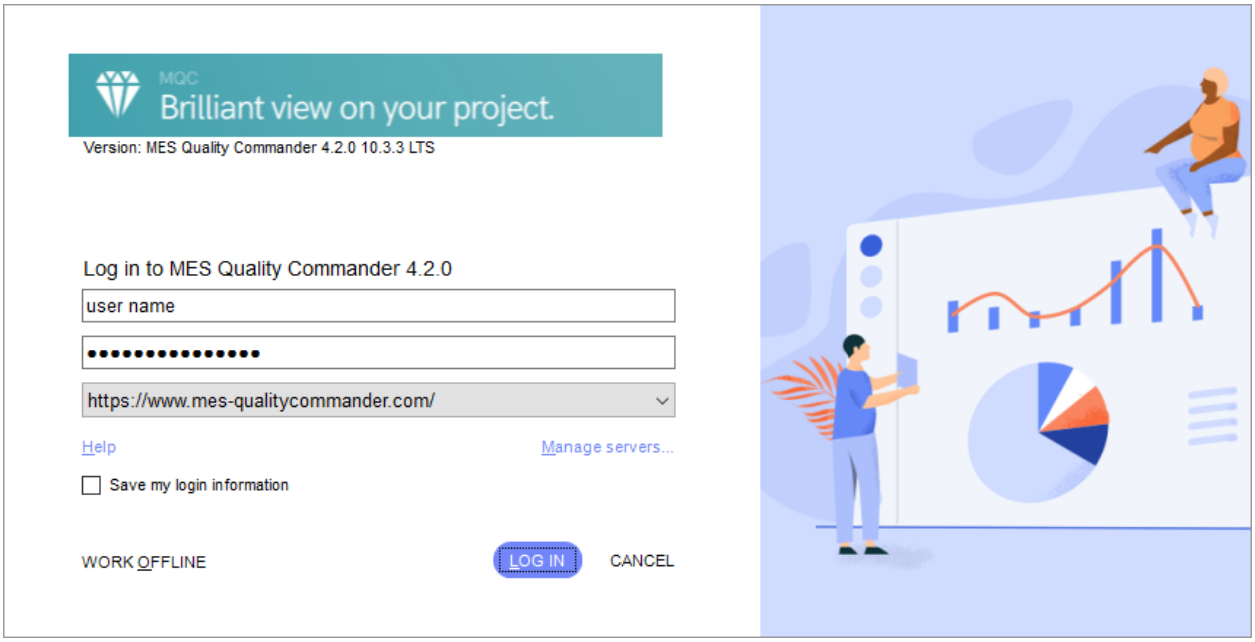


Figure 4.1: Start MQC and log in by entering server URL, user name, and password. Alternatively choose to work offline.

If you start MQC for the very first time, you must connect to the MQC server to receive or validate your license. Therefore, please enter the URL of your MQC Server (see Figure 4.1 above), user name and password

and then log in. This will automatically provide you with the latest updates, bug fixes and other improvements.

4.2 CREATE NEW ANALYSIS

To analyze and visualize your project data with MQC, first you have to create a new analysis. From the menu bar choose **Tools/New Analysis**. To be able to see the menu bar, please, close the **Files and data** dialog, which is preselected after starting MQC.

Note: Whenever you start a new analysis, please, make sure to save your old project before.

Once the new analysis has been created, open the **Configuration/Data Sources** dialog on the left-hand side panel to import your project data.

Select to add data from directories (see Figure 4.2). Then navigate to the folder where you have stored the previously extracted sample project data provided by MQC, choose the **Data** directory and commit your choice.

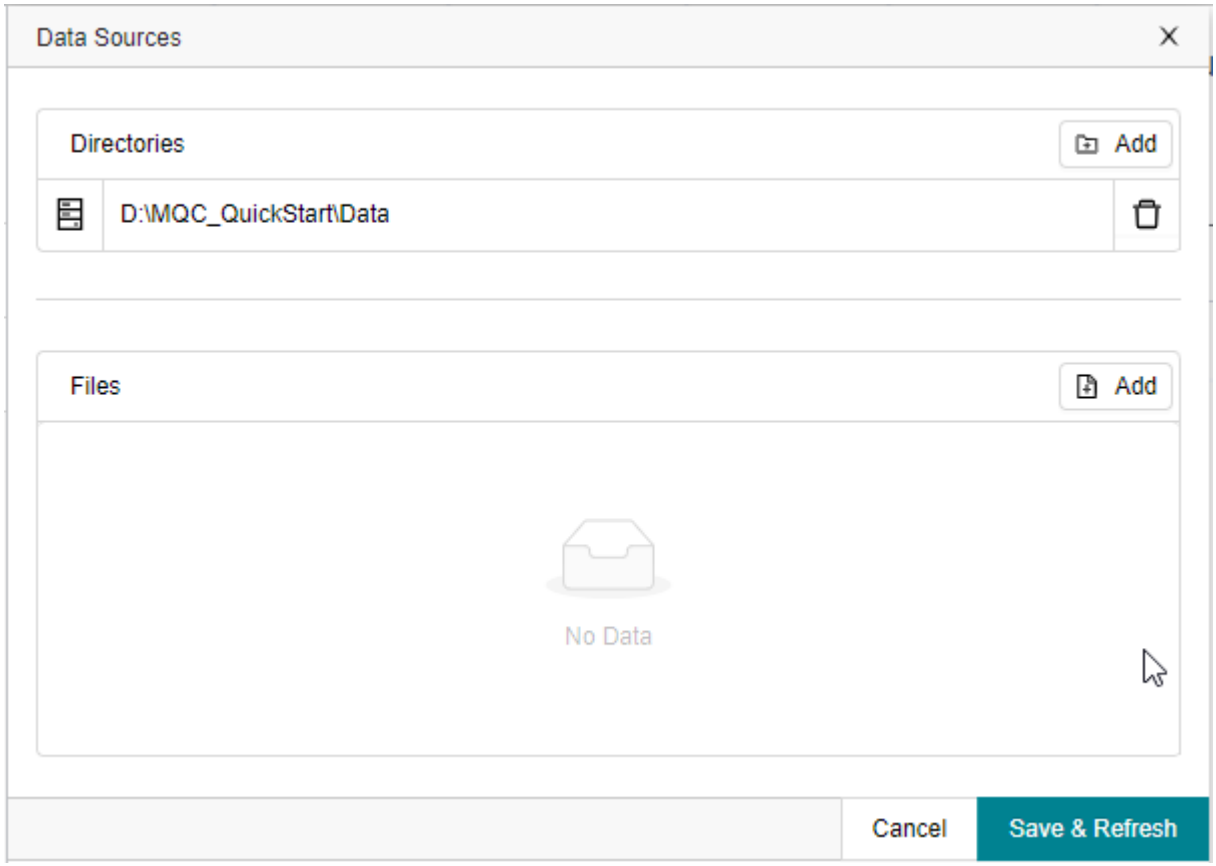


Figure 4.2: Import project sample data into MQC by adding the directory including the sample data

The directory (including all subdirectories) will be recursively searched for files that can be read by the MQC tool adapters (for detailed information on the supported adapters, please refer to *Data Sources*).

MQC now loads and transforms the data.

Changes inside the added directories (new files, removal of files or even changes within the files) are automatically detected and can be applied either manually or automatically if “Automatic Refresh” is enabled (see chapter *Actions and Configuration Menu*).

4.3 CHECK DATA AVAILABILITY

The Data Availability page gives you a first overview of the imported data. This page helps you to evaluate, which of the data you expect to be collected from a particular data source actually is available respectively missing. Availability is shown per artifact, per data source and for the whole project (see Figure 4.3).

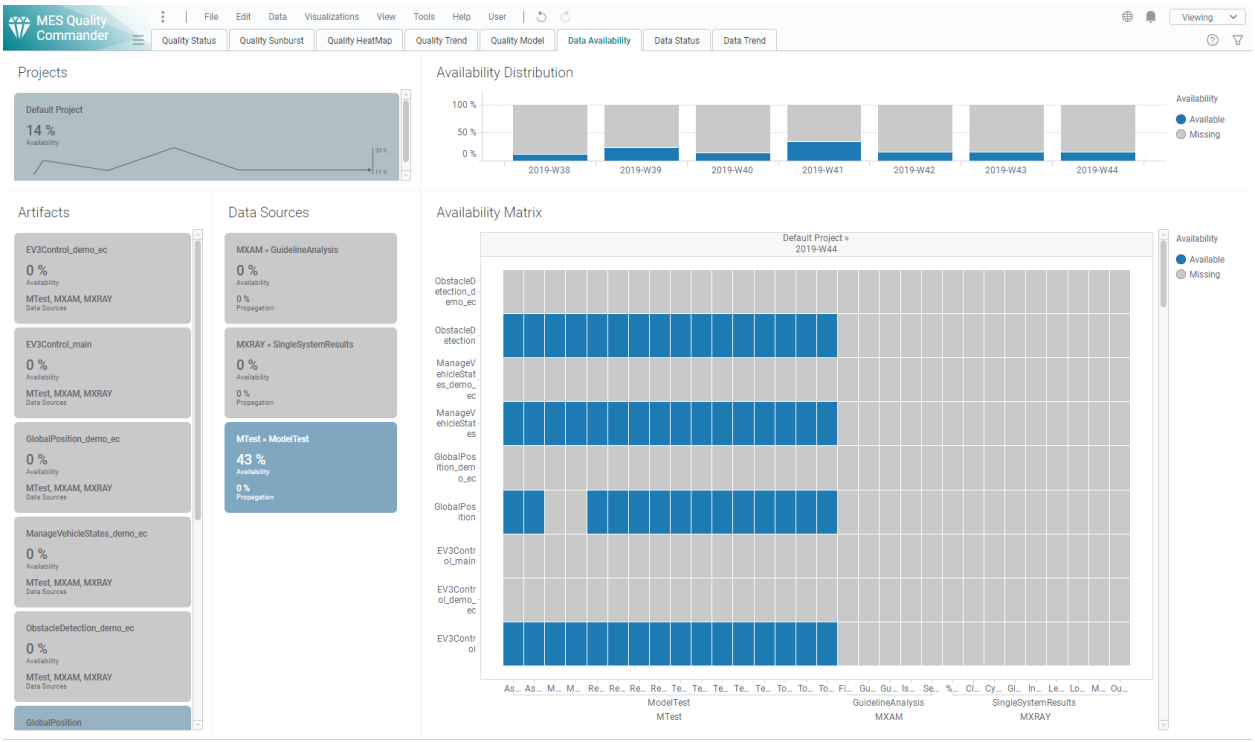


Figure 4.3: MQC Data Availability page showing the availability of the imported project sample data

Data availability can be checked per revision by scrolling from the last to the first revision in the *Availability Matrix*. Additionally, the *Availability Distribution* bin chart at the top of the page depicts for each revision the relative portions of available and missing data.

MQC uses a blue data coloring scheme on pages related to data. In context of the Data Availability page, this means you can easily distinguish between available data in blue and missing data in grey.

You may have noticed (by looking at the Availability Matrix visualization) that there are a couple of artifacts with similar names. Additionally you may detect that for a selected artifact there is only data available from one data source. Yet for other data sources there still appear grey (missing) tiles. In this particular example the reason for that is the current artifact naming, which can be adapted by loading some project configurations.

4.4 IMPORT PROJECT STRUCTURE

Until now, the names of the observed artifacts were taken as provided by the data sources itself. However, it may be the case that different tools collecting data and creating reports are using different names for the same artifact. Therefore, MQC provides configuration possibilities to map these different names to the same artifact name.

The import of configuration files can be done in the Configuration section of the menu in the bottom-left of the MQC working space. Per default the menu is enabled on all MQC pages. You can simply switch it on or off by clicking on the `Menu` icon within the MES Quality Commander banner in the top-left.

Figure 4.4: Enable the menu within the MES Quality Commander banner

From the left-hand side panel choose **Configuration/Project Structures**, click on the `Load` button and navigate to the folder, where you have stored the sample project data. Choose the Project Structures Excel file `01_ProjectStructures_Initial.xlsx` and commit your choice.

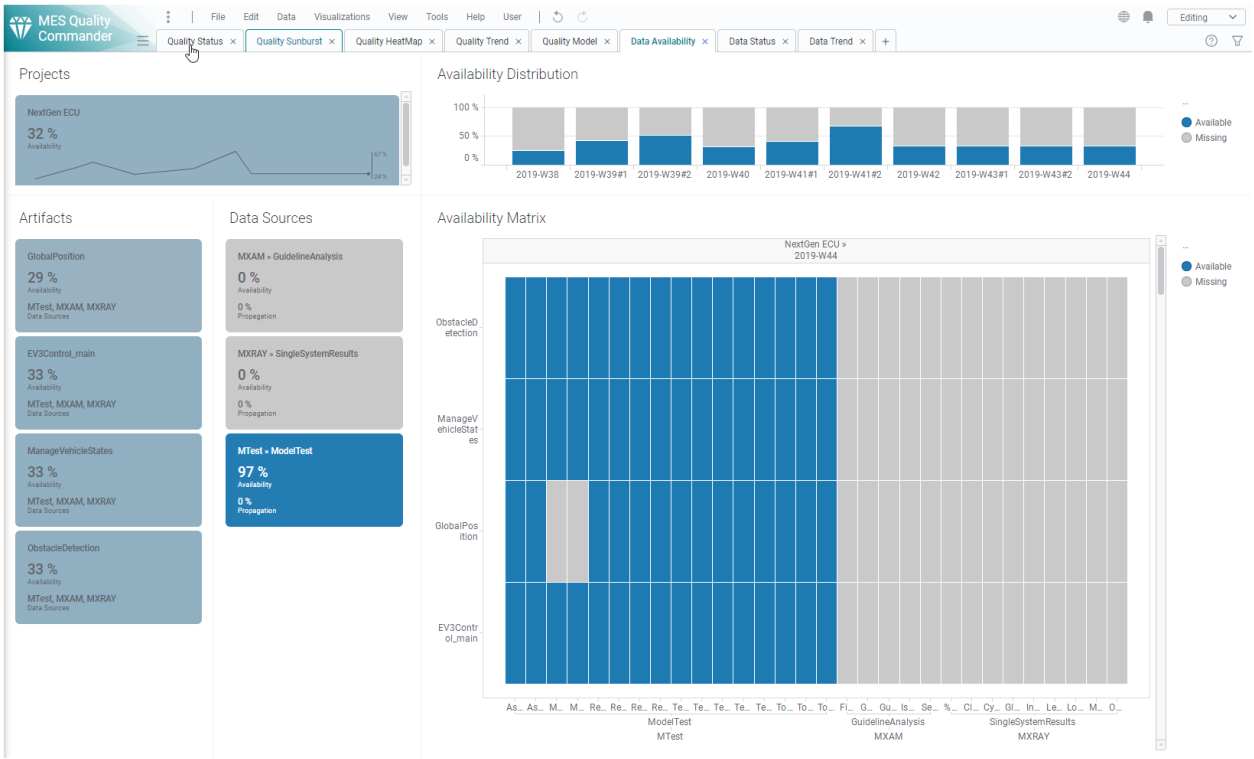


Figure 4.5: MQC Data Availability page after importing project structures

As you can see, the number of artifacts shown on the left-hand side has been reduced. All loaded measure values are automatically assigned to the correct artifact according to the imported mapping defined in the Sample Project Structures file.

Additionally a dedicated name was chosen for the project, which now replaces the previously used "Default Project" and that is to be seen on the Project KPI tile in the top-left corner of the page.

For more details on structures and dimensions, refer to *Dimensions and Structures*. For details on how structures and dimensions can be configured in MQC, refer to *Project Structures Configuration*.

Even after loading a project structure containing a proper artifact mapping, you still see missing data. This is related to the fact that data always is assigned to the revision that matches the point in time the data was measured.

Please note, that if for a particular revision a specific measure value is missing but was imported for any of the previous revisions, MQC offers a propagation feature. On the left-hand side panel open the **Configuration/Settings** dialog. In the **Propagation** section select "All Data" and keep the time range as is. Then click the `Update` button in the bottom-right corner of the dialog to submit your changes. This will copy previously collected data to later revisions to gain a full set of available information (refer to *Data Propagation* for more details).

Now you can directly check the Data Status page.

4.5 CHECK DATA STATUS

On the Data Status page you will find all available measure values for each of the observed artifacts.

To restrict your data only to the interesting information, you have two possibilities. On the one hand, you can select elements (in MQC called marking). When marking one or multiple elements, data that corresponds to elements not marked is excluded from all visualizations. Please note that marking works cumulative. That means further marking without reset will again reduce the result set of the last selection. You can always reset your selections by clicking on the *Project* KPI tile.

On the other hand, you can apply filters for quick data discovery using the filter panel on the right-hand side of the page. All visualizations, bar charts and KPI's will adapt to the selected filters. Remove the chosen filters by clicking the **Edit/Reset visible filters** from the menu bar.

4.6 IMPORT QUALITY MODEL

Before analyzing the quality that is computed from the available data, you should first import a Quality Model that fits to your collected measures and that defines how quality should be calculated for your project.

From the left-hand side panel choose **Configuration/Quality Model**, click on the `Load` button and navigate to the folder, where you have stored the sample project data. Choose the Quality Model Excel file `02_QualityModel_Initial.xlsx` provided by MQC and commit your choice.

Typically, a Quality Model defines a number of quality properties and corresponding measurement functions, which use your imported project data to calculate quality values for the defined properties. For more details, please refer to the explanation of *Quality Concept* and *Quality Model Configuration*, respectively.

After you have finished importing the Quality Model, MQC calculates quality. You may now check the Quality Trend page.

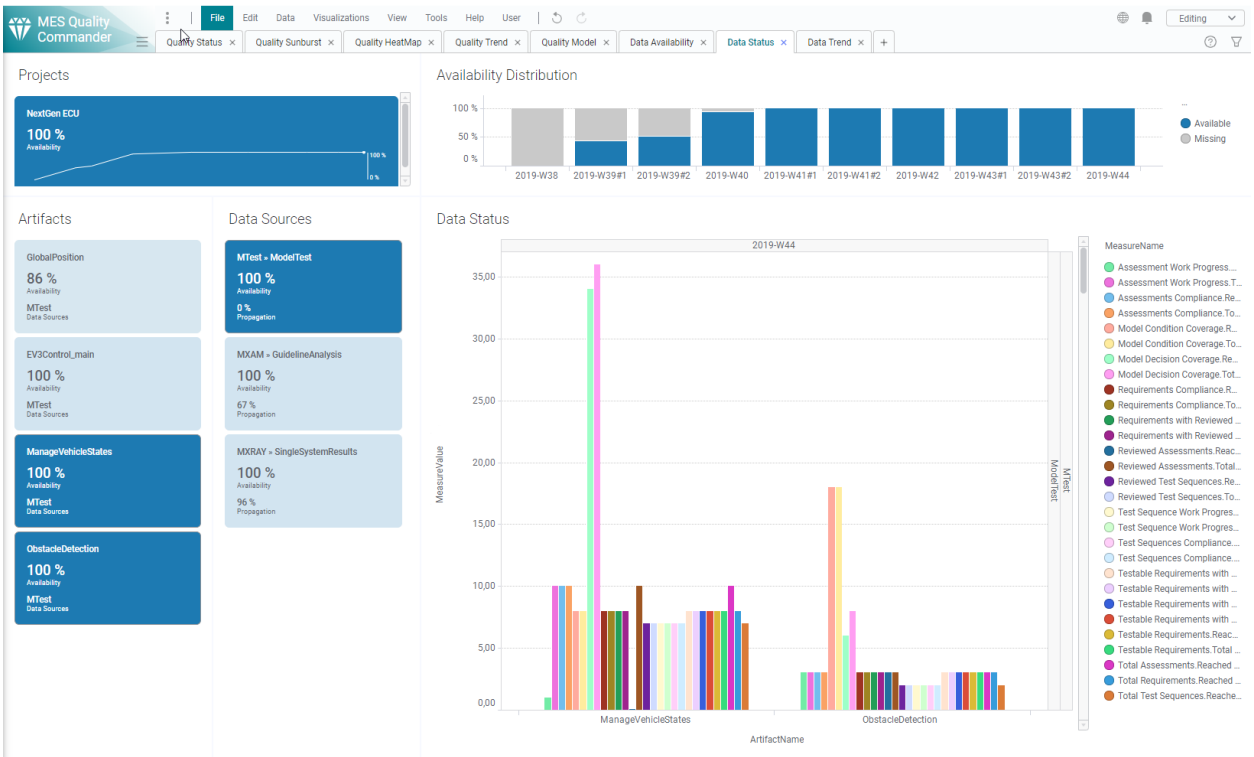


Figure 4.6: Using Marking on MQC Data Status page: The main visualization shows all base measures related to the selected Artifacts and Data Source (MTest)

4.7 CHECK QUALITY TREND

The Quality Trend page gives you an overview on how the quality of your project has evolved over time. A quality trend is shown for each of the observed artifacts as well as for every quality property defined by the imported quality model.

The quality for each quality property is calculated from the imported measure values using a measurement function defined in your quality model. For more information, please refer to the definition of *Quality Properties*.

A *Quality Bin Distribution* chart depicts for each revision the relative portions of quality properties with good, acceptable and bad quality.

Quality is visualized using a traffic light coloring scheme, i.e. green for good quality and red if the computed quality is bad. Therefore, you can easily detect problems, i.e. those parts of your project with insufficient quality.

To focus on specific information only, you may select on the left-hand side KPIs a single or even multiple artifacts as well as one or more specific quality properties. Thus, the number of trend lines shown in the main visualization is reduced based on your selection.

Still, each artifact trend line shows an aggregated result based on all contributing quality property measure values. The same applies to each quality property trend line.

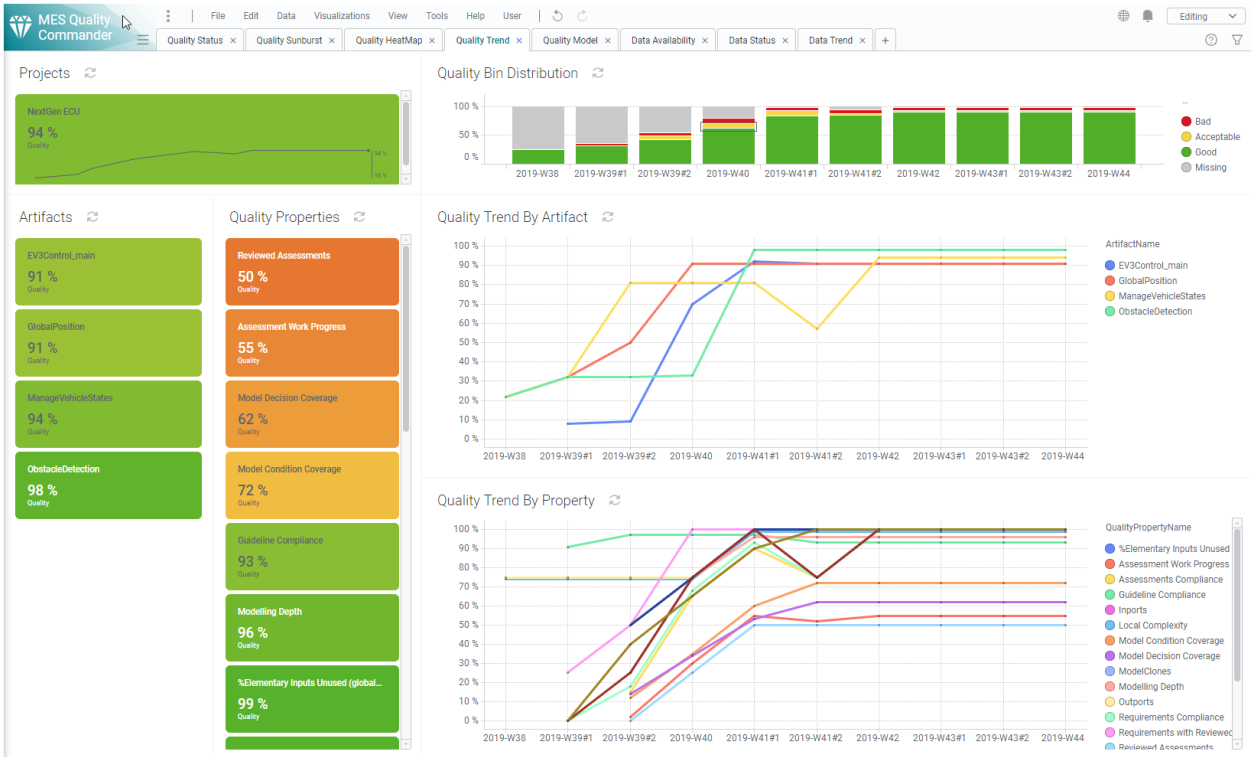


Figure 4.7: MQC Quality Trend page showing the evolution of project quality over time based on revisions

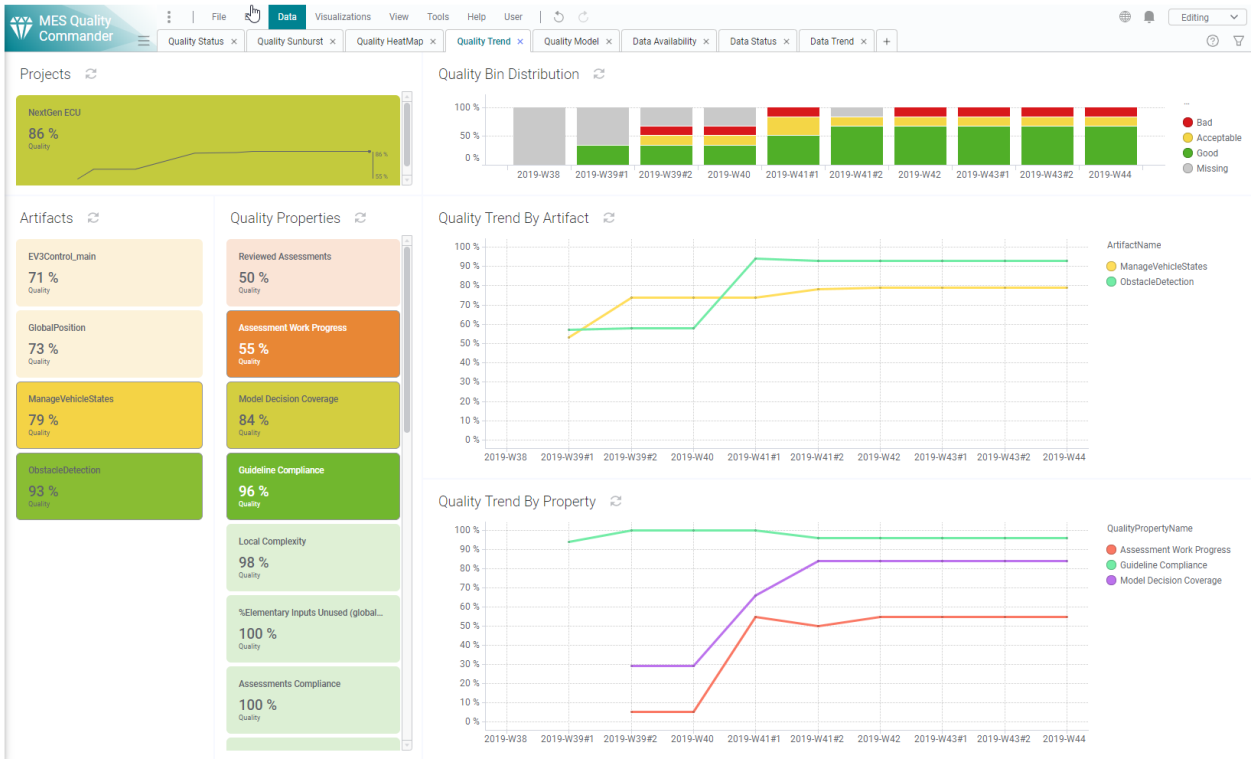


Figure 4.8: MQC Quality Trend page just showing Artifact and Quality Property Selection

If you are interested in quality per property and artifact, please switch to the Quality Status page.

4.8 CHECK QUALITY STATUS

The Quality Status page provides you detailed information on the calculated quality. Here you can check quality for each of the artifacts separately resp. for each quality property per artifact at a certain point in time.

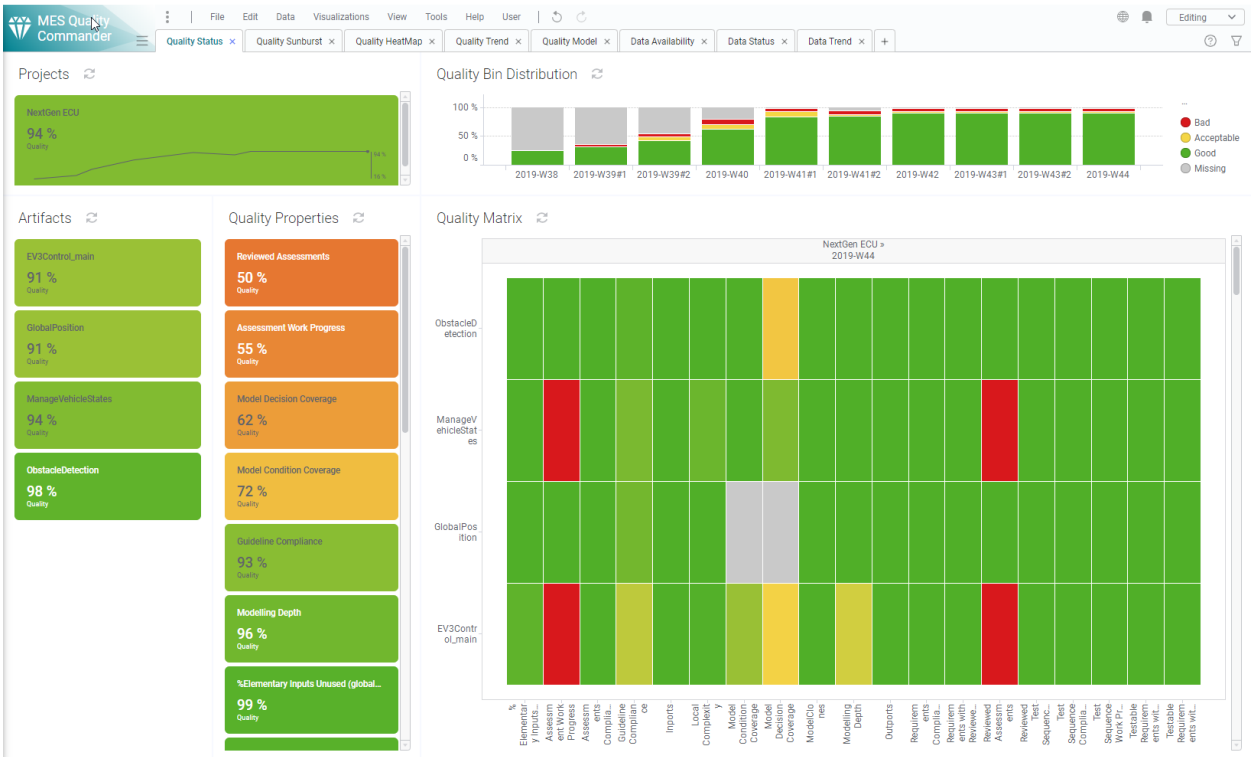


Figure 4.9: MQC Quality Status page showing detailed quality information per quality property and artifact

The quality matrix shown in the main visualization window of this page can be used to gain information about a particular quality property for a certain artifact.

Grey areas inside the *Quality Matrix* indicate missing quality properties, which means the data used to calculate the quality property is not available for that particular revision.

4.9 WORK WITH MQC

Following the steps described in the previous sections, you should gain a general overview about the main MQC pages and visualizations.

As you may have noticed, all pages use a common layout. A *Project* KPI on the top-left always informs you about the overall quality respectively data availability of your project including a trend visualizing the progress over the past revisions. On the right-hand side of the *Project* KPI, you will always find a revision

based bin chart, where each bin shows relative portions of certain sets of elements compared to other sets, i.e. available data vs. missing data or good quality items vs. bad quality items.

On the left-hand side a list of selectors is available, which enables you to reduce the displayed information, e.g. by selecting one or multiple artifacts, quality properties or data sources.

Each page contains a main visualization window, where you can find all necessary information regarding project quality resp. data availability, and which always reflects your selections previously done.

You can now start identifying the reason for bad quality. The general concept of detecting issues is to click on (or hover over) e.g. red quality bins or tiles. At the Quality Status page choose the last revision from the *Quality Bin Distribution* chart at the top and select the legend entries “Bad” and “Acceptable”.

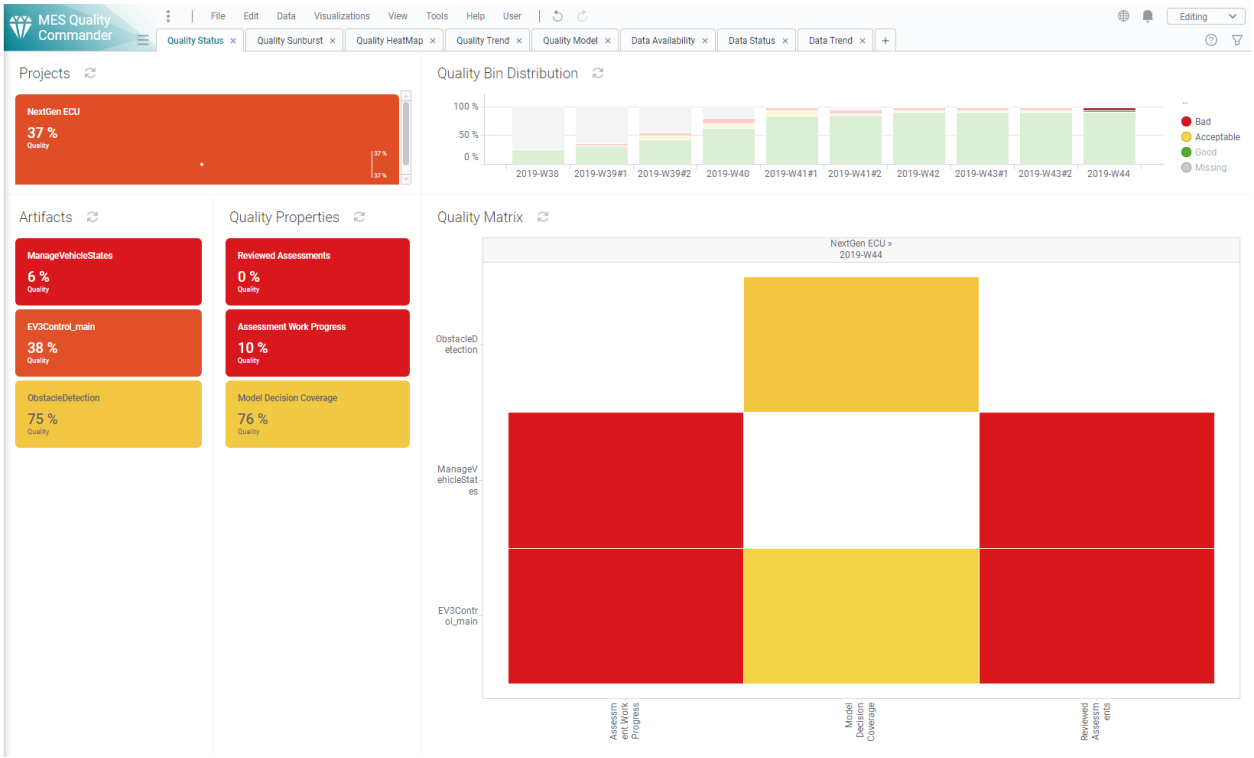


Figure 4.10: MQC Quality Status page after marking to reduce the main visualization to quality properties with bad and acceptable quality

The main visualization window immediately shows you all quality properties with an insufficient quality per artifact. By hovering over the tiles, you will get detailed information about particular quality properties, for example the concrete quality measure value.

Select a particular (or various) tile(s) inside the *Quality Matrix*. Then click on the icon within the **Data of the Quality** menu of the left-hand side panel as shown in Figure 4.11.

MQC directly switches to the Data Status page and reduces the visualization to those base and derived measures that were used to calculate the selected quality properties within the chosen revision.

Please note, this selection is kept when you now navigate between the data pages to get all necessary information related to the selected data, e.g if you now switch to the Data Trend page.

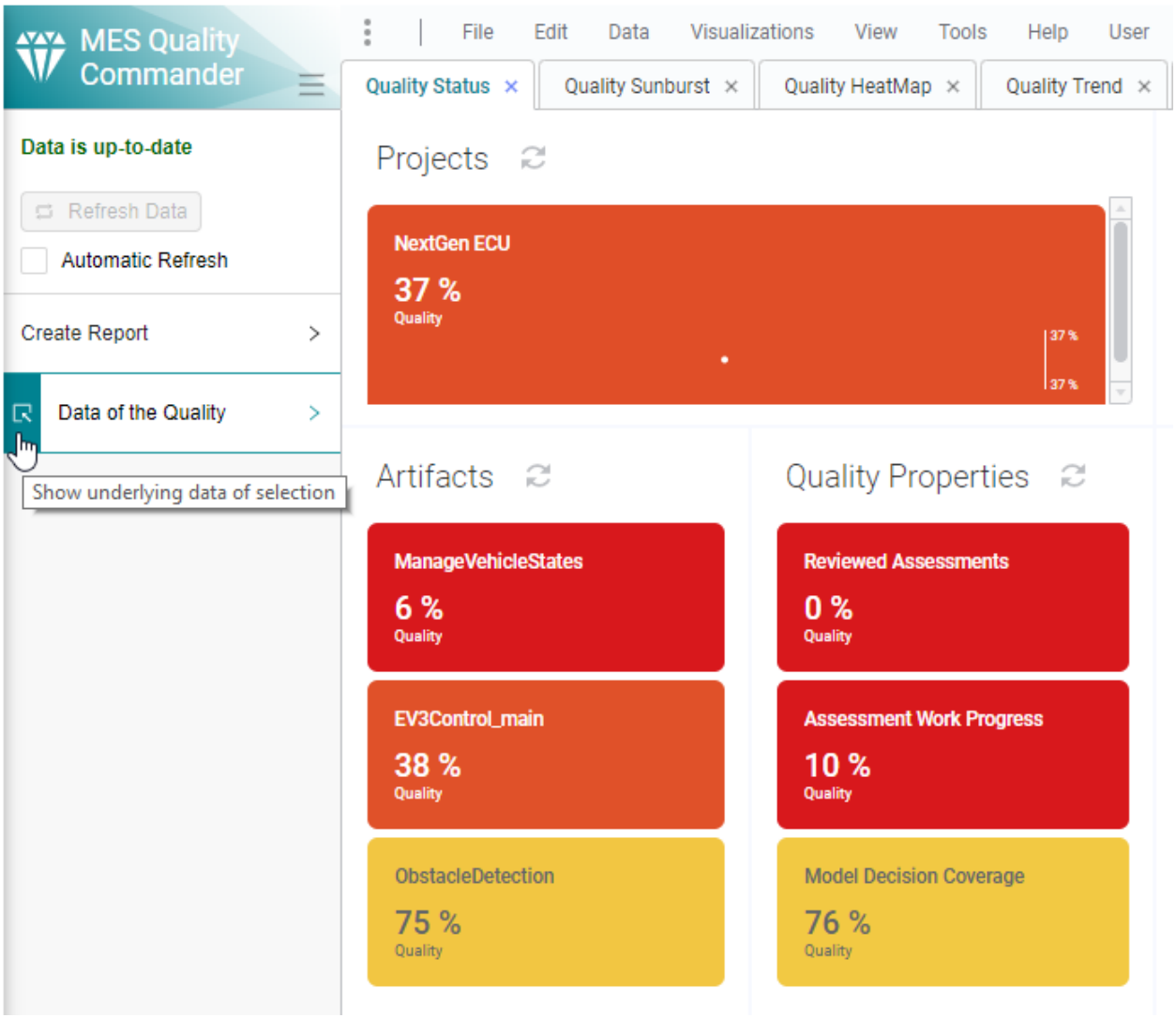


Figure 4.11: Show Details for Data that is used to calculate Quality Properties with insufficent quality

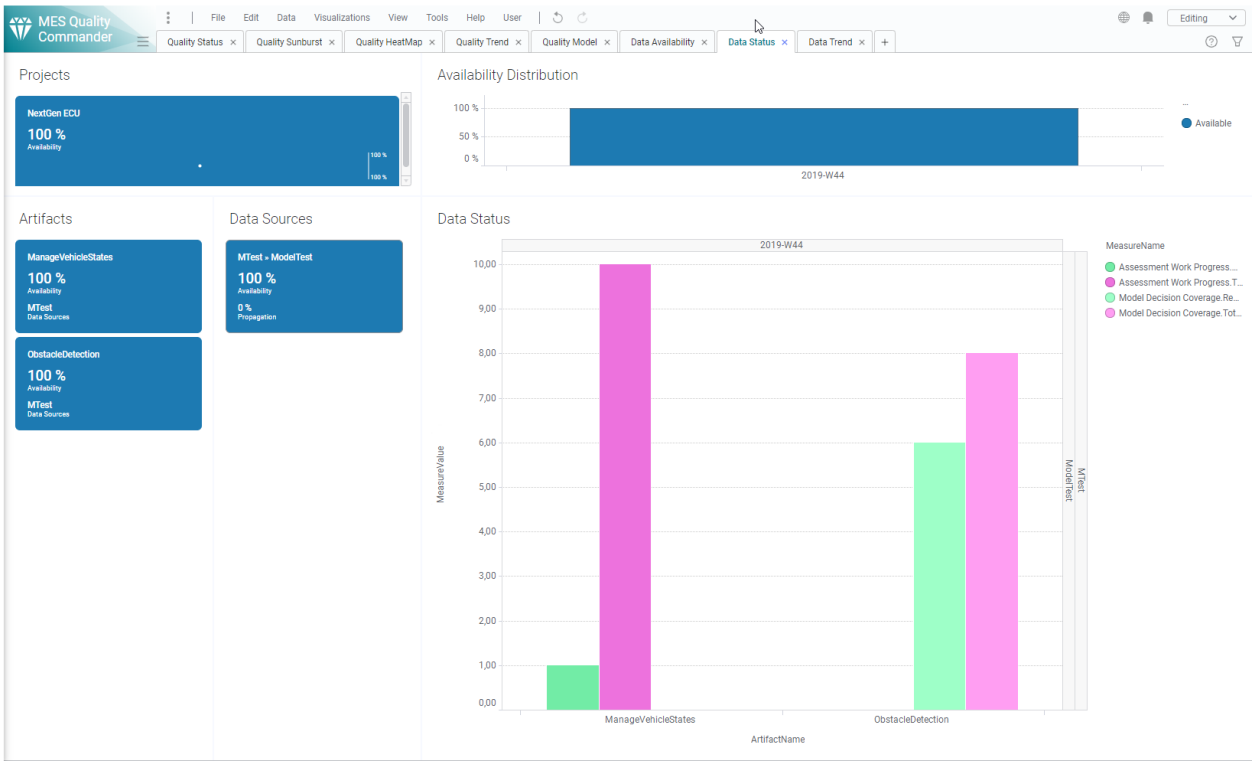


Figure 4.12: MQC Data Details View showing Base and Derived Measures used to calculate selected Quality Properties

For more details on the MQC Drill Down functionality, please refer to *Data Details Drill Down*.

At the end, with the possibilities offered by MQC, you can easily identify those parts of your project having problems and which may require further activities to improve quality.

5 DIMENSIONS AND STRUCTURES

The aim of MQC is to compute quality for a project based on provided data and to visualize this data in a structured way. In order to do this, MQC structures the provided data systematically. This is done with the help of artifacts, projects, revisions, and milestones.

Based on this structuring, MQC can offer several views on the provided data as well as the derived quality. MQC follows the terminology of ISO 25010, ISO 25012, and ISO 25022 when structuring the provided data and the computed quality. This chapter gives an overview of the terms used in MQC and their relation to ISO 25010, ISO 25012, and ISO 25022.

5.1 ARTIFACTS AND ARTIFACT STRUCTURE

When data is collected within MQC, it is linked to objects for which that data was measured and which should be used for visualization and quality computation. Artifacts in MQC are any objects for which quantitative data is collected. Examples of artifacts are Simulink models, generated code or documents like review protocols. Figure 5.1 shows examples of artifacts. The Simulink model, the requirements document, the respective TargetLink model, and the generated code all represent artifacts.

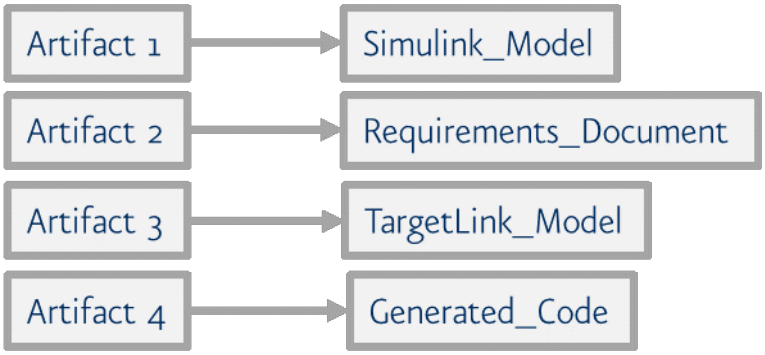


Figure 5.1: Example of artifacts in MQC

However, artifacts can also be defined in a more general sense. They do not always need to represent real objects as work products. Artifacts can also represent virtual objects like process steps, for example.

Artifacts can be grouped hierarchically. This structure can represent a state of the artifact itself, but it can also represent a logical structure. For instance, Simulink models can be grouped based on their functionality. However, a Simulink model can also be grouped with respect to different responsibilities (e.g. model 1 and 2 are grouped by role 1). As an example, consider several Simulink and TargetLink models that are

grouped according to their implementation. See Figure 5.2 for an example of the general artifact grouping in MQC.

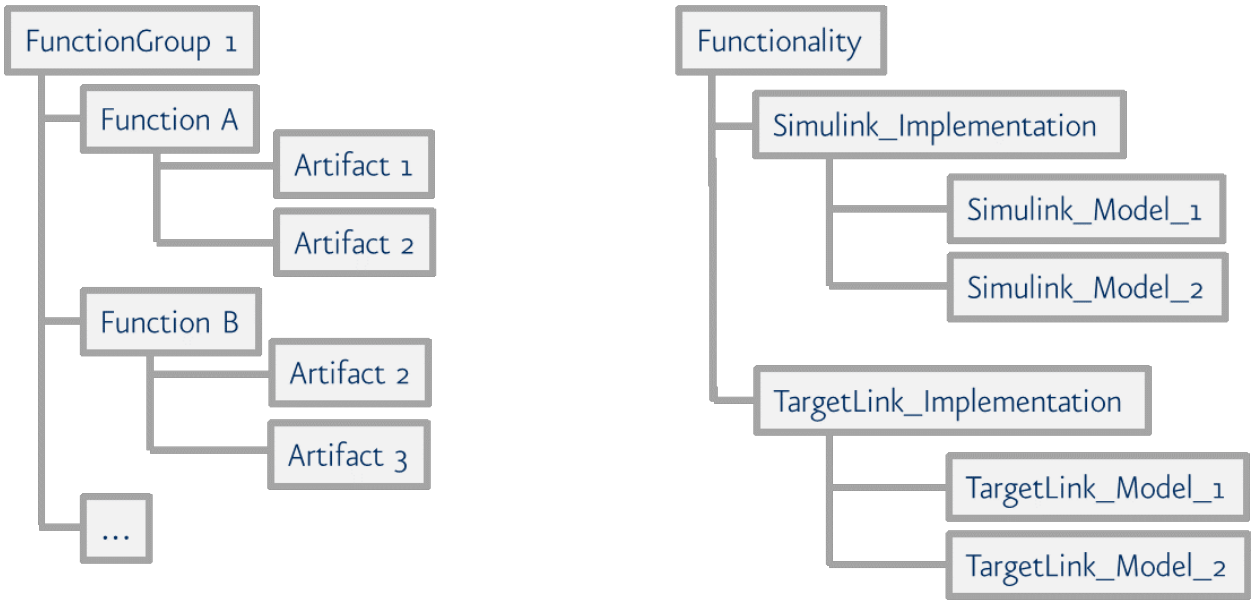


Figure 5.2: Artifact grouping in MQC

For details about how artifacts can be grouped in MQC, please refer to *Artifact Structure*.

5.2 REVISIONS

In real-world projects, data is collected for artifacts at different points in time. When data is imported into MQC, it is assigned to revisions. Assigning the data to revisions depends on the time stamp that is provided by the data source. If there are multiple instances of data available for the same revision, usually the latest data instance is used and visualized in MQC. As an example, consider a weekly revision granularity and several reports with data created on different days of a week. Then, the one with the latest time stamp is used for the weekly revision, see Figure 5.3

MQC currently supports three types of revision granularities: daily, weekly and monthly .

5.3 PROJECTS AND MILESTONES

In MQC, projects can be defined in order to represent real world development projects. The artifacts with the given data will be linked to these projects. Typically, one or multiple artifacts belong to a certain project. Nevertheless, please note that one artifact can be linked to several projects, too. However, the artifact structure as described in *Artifacts and Artifact Structure* is always defined for an artifact which is independent of a certain project. The relationship between artifacts and projects and the artifact structure is shown in Figure 5.4.

Defining relations between your artifacts and certain data sources, measurements, measures or even variables can be helpful to configure your project properly. You can assign data sources, measurements or

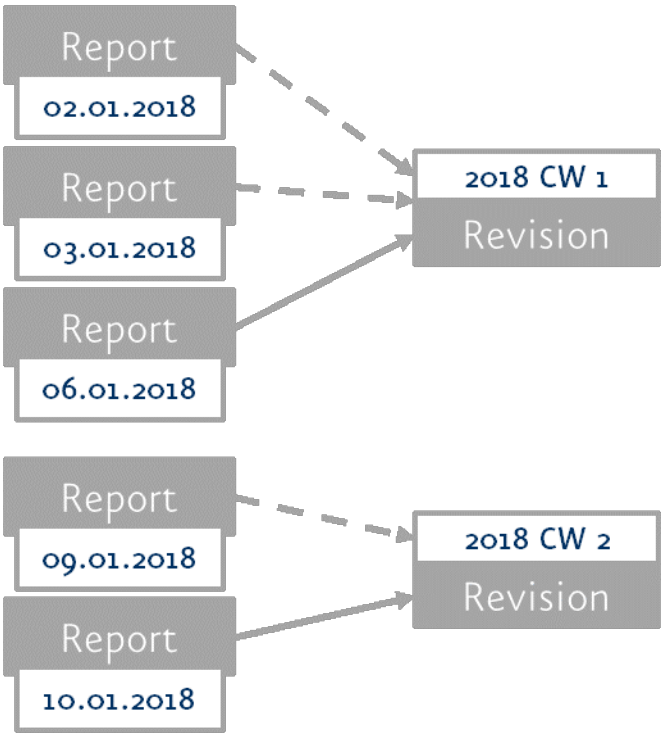


Figure 5.3: Assigning data to revisions

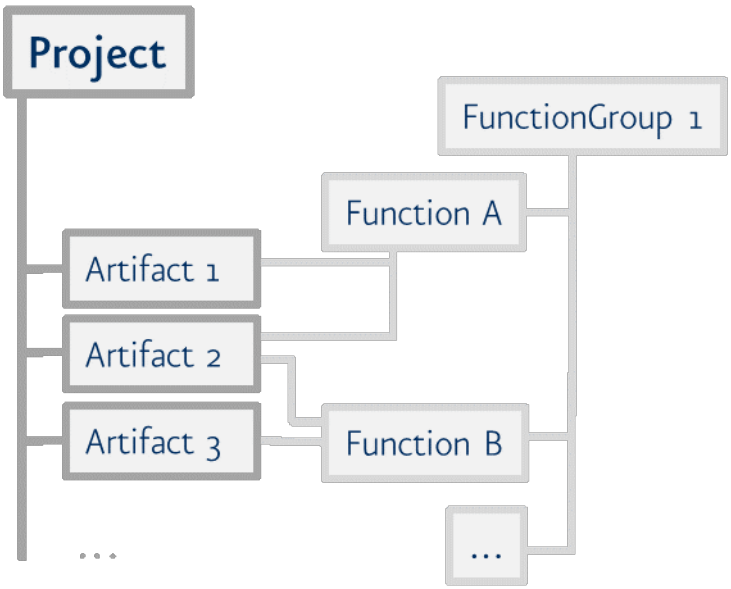


Figure 5.4: Relationship between artifacts and projects

measures to certain artifacts, defining the exact data that is expected ("whitelisted") or excluded ("black-listed") for these artifacts within your project. You can define that association between Data Sources and Artifacts via Context Categories. The following figure shows the relationship between artifacts and data sources (including measurements and measures).

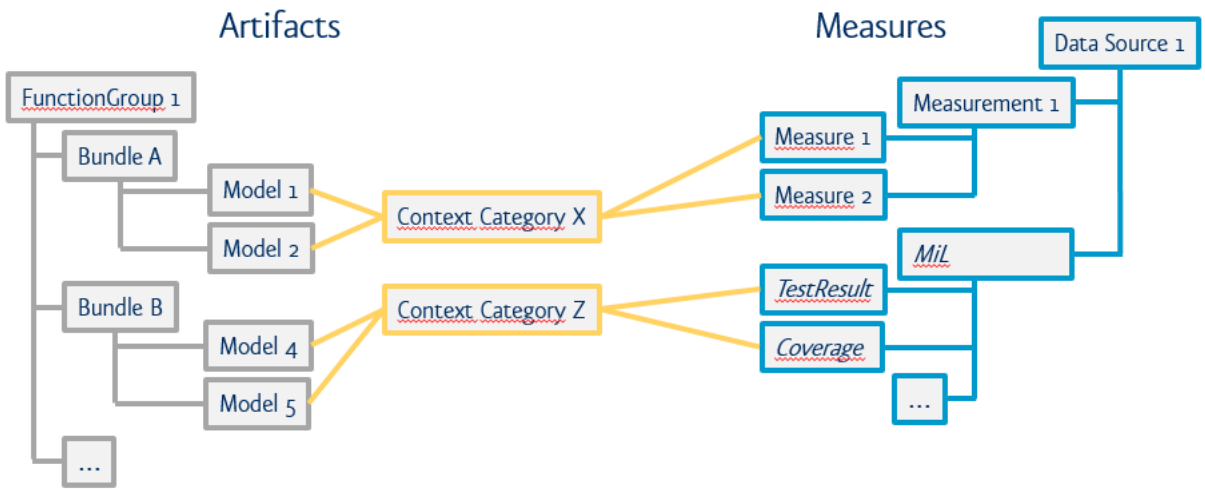


Figure 5.5: Relationship between artifacts and data sources

For more information concerning the usage of Context Categories for your project, please refer to the section with the same name (see *Context Categories*).

Each project can define its own milestones, where a milestone represents an important development step, i.e. the milestone time defines the end of a respective period, which starts at the end of the previous milestone and lasts until the current milestone time.

In MQC, this is realized by clustering revisions and linking the clustered revisions to a milestone defined for a certain project according to a common timeline. This means all revisions with a start time after the previous milestone and before the current milestone are linked to the current milestone. With that, data assigned to a revision is also linked to the milestone, to which the revision belongs.

The relationship between projects, revisions and milestones is depicted in Figure 5.6.

For details on how to define projects and their milestones in MQC, please refer to *Project Naming* and *Project Milestone Structure*, respectively.

5.4 MEASURES AND MEASUREMENTS

Data is collected from a source via measurements. A measurement yields one or multiple measures. This is done for an artifact at a specific point in time (by a revision). Figure 5.7 shows this workflow, i.e. the measurement yields data represented by the three dimensions Measure, Artifact and Revision.

In the following, only one artifact for a fixed revision is considered so that the dimensions of the data cube are reduced to one, see Figure 5.8.

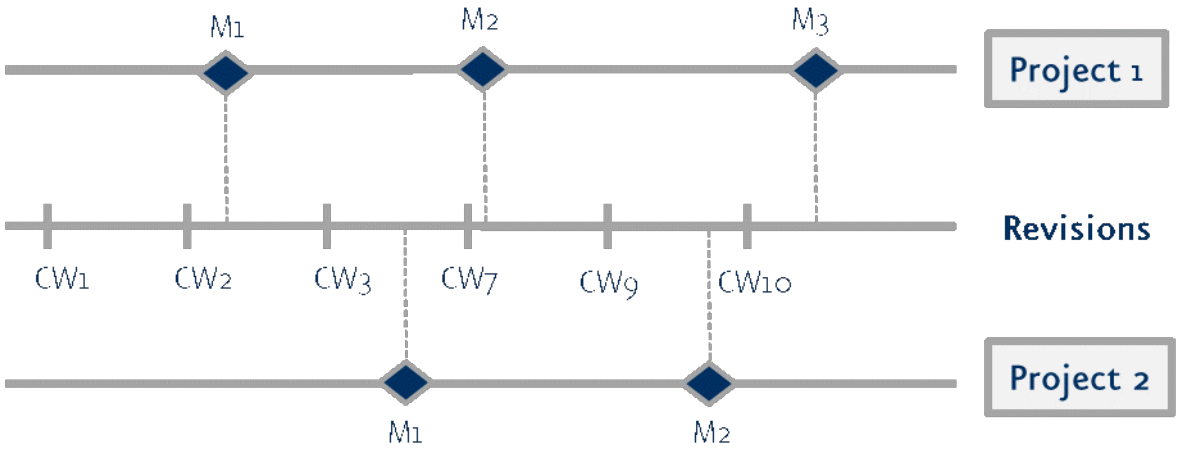


Figure 5.6: Relationship between projects, their milestones and revisions

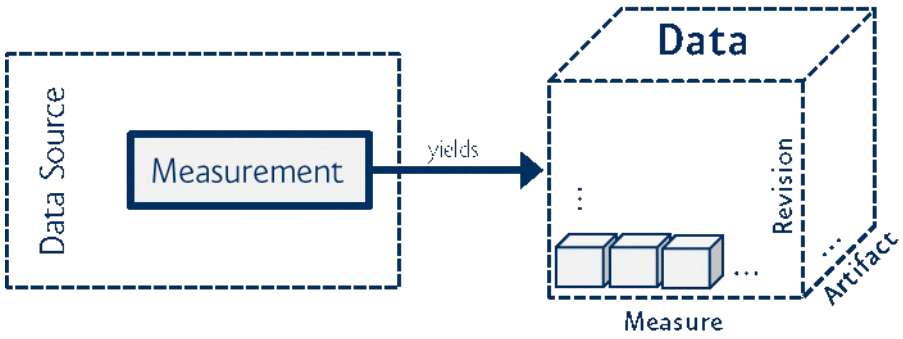


Figure 5.7: From measurement to data

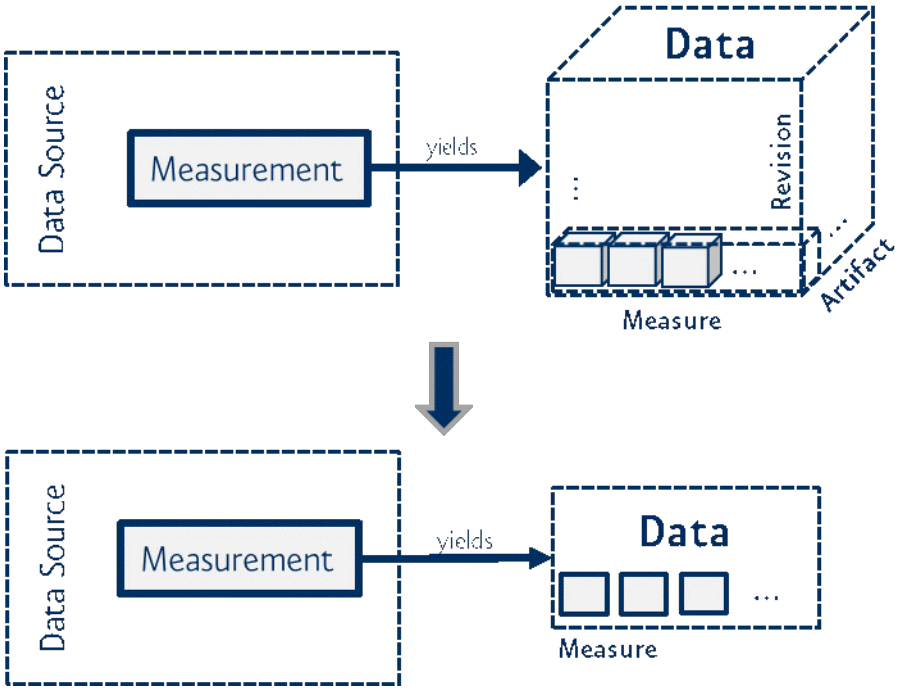


Figure 5.8: Freeze artifact and revision dimension

The measurement is the process, by which data is collected and by which the values of one or multiple measures are determined. Therefore, a measure is a variable to which a value is assigned by the execution of a measurement (see Figure 5.9).

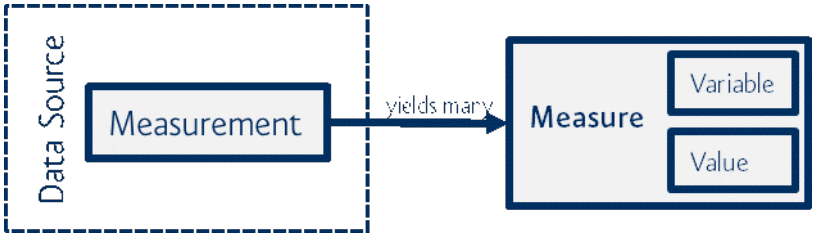


Figure 5.9: Measurement yields measures

An example is the measurement “Measuring MISRA compliance” for model A and revision X, which results in a guideline report with, for example, the measure variable Passed and the measure value 10.

MQC often imports data from tools directly. The source of the data is named data source. For example, the tools MXAM, M-XRAY or MTEST can be data sources.

A measure that originates directly from a data source is called Base Measure.

Please note that there is no one-to-one relation between measurement and data source. Rather it could be that multiple measurements are relevant for the same data source. For example, when using MXAM as a data source, the measurements “Measuring MISRA compliance”, “Measuring MAAB compliance” and “Measuring TargetLink Known problems compliance” might be considered. Each of these measurements corresponds to one guideline document, for which compliance is checked, and results in one or multiple

measures, where each measure consists of a measure variable and a measure value. For example a measure variable Passed may not only exist for the measurement "Measuring MISRA compliance", it might also exist for "Measuring MAAB compliance" and "Measuring TargetLink Known problems compliance". See Figure 5.10 for the relationship between data source, measurements and base measures.

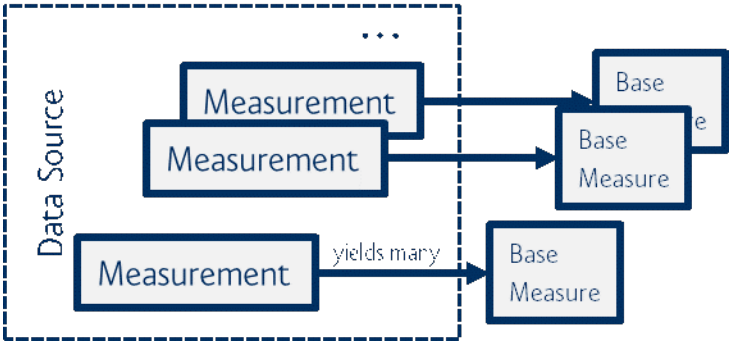


Figure 5.10: Data Sources, measurements and base measures

In MQC, measures can be grouped by type. As an example, consider the measurement "Measuring Local Complexity" for the data source M-XRAY. This measurement yields the base measures `Local Complexity Good`, `Local Complexity Acceptable` and `Local Complexity Bad`. These base measures can be grouped into a base measure called `Local Complexity`, which then contains three elements, each of the elements consisting of a measure variable (i.e. `Good`, `Acceptable` and `Bad`) and a corresponding measure value.

To address the elements of a base measure group, the notation BaseMeasure.BaseMeasureElement is used.

5.5 DERIVED MEASURES

MQC differentiates between base measures (see *Measures and Measurements*) and derived measures. Derived measures are computed either from base measures or from other derived measures (see Figure 5.11). They are used to visualize their trend and/or to simplify the quality computation.

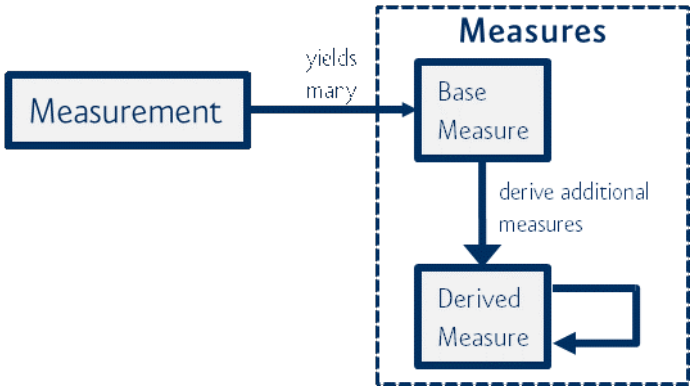


Figure 5.11: Relationship between measurements, base measures and derived measures

As an example, consider the measurement "Measuring MISRA compliance" with the base measure variables `Passed`, `Failed` and `Warning`. Here, a suitable derived measure variable `Total` could be added, which stands for the total number of all `Passed`, `Failed` and `Warning` guidelines (see Table 5.1):

Total = Passed + Failed + Warning

Table 5.1: Derived measure for measuring MISRA compliance

Measurement	Base Measure			Derived Measure
	Passed	Failed	Warning	Total = Passed + Failed + Warning
Measuring MISRA Compliance	10	5	5	20

The function for the derived measures can be used for multiple measurements as long as they have the same base measures. For the above example, this means the derived measure Total can also be computed for the measurements "Measuring MAAB compliance" and "Measuring TargetLink Known problems compliance".

For details how derived measures are defined in MQC, please refer to *Derived Measures*.

6 QUALITY CONCEPT

This chapter explains how MQC computes quality based on the imported data, derived measures (both are introduced in *Derived Measures*) and a Quality Model. Additionally it describes how quality is structured in MQC.

6.1 QUALITY COMPUTATION

When data is collected and imported in MQC, it is handled in terms of measures. In the following, we consider one artifact and one revision.

A measurement – that is the set of operations executed to determine values for measures – is applied and it yields a collection of measures including their measure values. Either these measures can be base measures coming directly from the data source or derived measures, i.e. computed from base measures or other derived measures (see Figure 5.11).

Two or more of these base and derived measures are used to calculate a quality measure between 0 and 1. This quality measure is associated to a so-called quality property. Quality properties define the lowest level of computed quality. Examples of quality properties are "Guideline Compliance" or "Test Sequence Compliance".

For each quality property, a measurement function defines how to calculate the quality measure by using base and derived measures.

Note: Quality measures always need to be between 0 and 1. This is ensured by MQC even if the measurement function yields a value outside these boundaries.

The workflow from measuring the data to calculating the quality measure for a quality property is given in Figure 6.1.

Please note that the quality measure value of a quality property can also be calculated by using more than one measurement. In this case, the measurement function to compute the quality measure is used for all the measurements from the same data source. Afterwards the results are aggregated to an overall quality measure value. One example is the quality property "Guideline Compliance", where for the measurements "Measuring MISRA compliance", "Measuring MAAB compliance" and "Measuring TargetLink Known problems compliance", three quality measures based on the same measurement function are computed, see Figure 6.2.

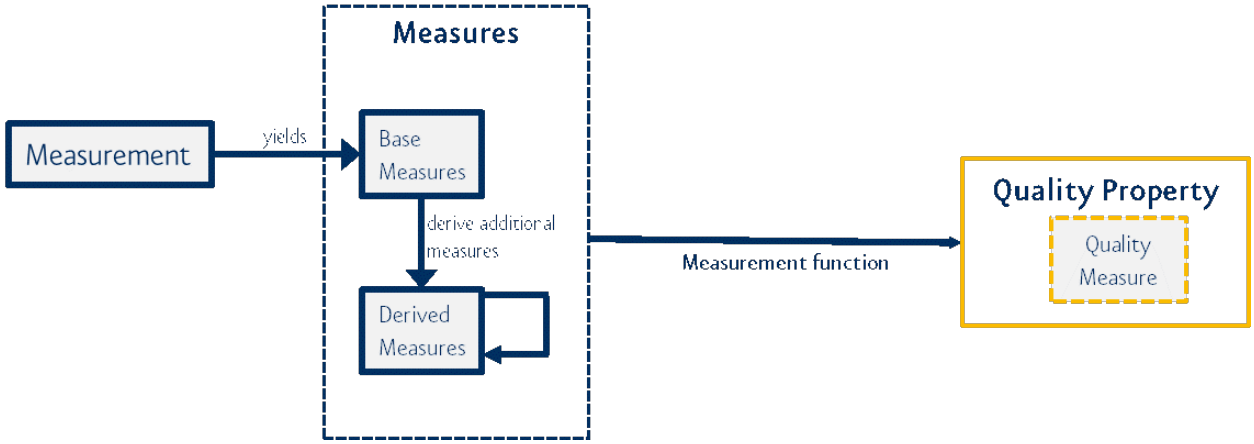


Figure 6.1: Workflow from collecting data with a measurement to calculating a quality measure by a measurement function

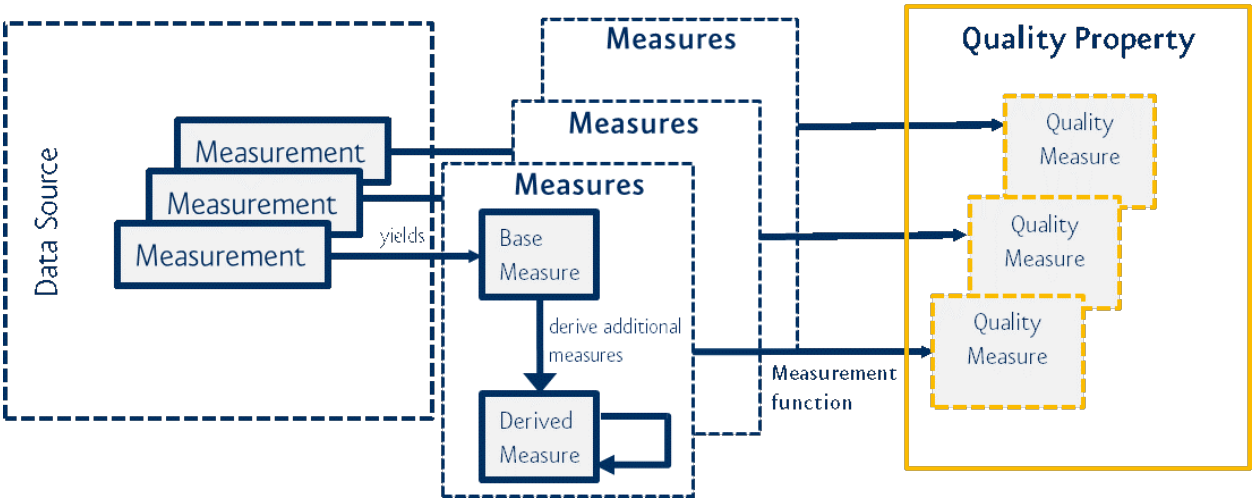


Figure 6.2: Measurements from the same data source use the same measurement function for computing the quality measures

An example for a simple measurement function is the computation of a quality measure value for the quality property "Model Decision Coverage" using the base measure values `Reached` and `Total`:

> [Model Decision Coverage.Reached] / [Model Decision Coverage.Total]

A more complex measurement function calculates the "Local Complexity" using the three different base measure variables `Good`, `Acceptable` and `Bad`:

> 1.0*[Local Complexity.Good] + 0.8*[Local Complexity.Acceptable] + 0.2*[Local Complexity.Bad] / [Local Complexity.Good] + [Local Complexity.Acceptable] + [Local Complexity.Bad]

The given base measures are weighted accordingly to compute a normalized quality value, i.e. between 0 and 1.

Please refer to *Quality Properties* for details on how to configure quality properties and measurement functions in MQC.

6.2 QUALITY BINS

Quality measures can be mapped to quality bins. A quality bin then contains a set of quality measures of a certain value range. The value ranges of the quality bins can be adjusted. Currently MQC uses a quality bin definition as follows:

- Bad, for quality measures in [0%, 20%]
- Acceptable, for quality measures in [20%, 80%]
- Good, for quality measures in [80%, 100%].

Next to the quality measure, the quality bin adds an additional attribute to each quality property (see Figure 6.3).

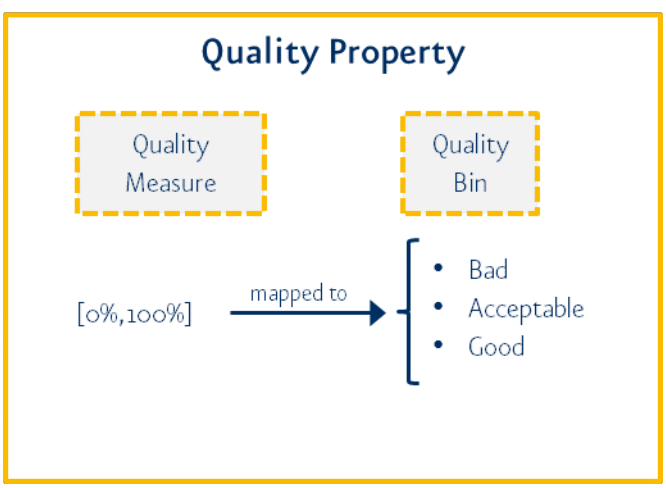


Figure 6.3: Quality bin as additional attribute in quality property

By counting the number of elements in the bin categories "Good", "Acceptable" and "Bad", MQC is able to provide a quality value distribution.

6.3 QUALITY MODEL STRUCTURE

Quality properties are the computable atomic elements of the quality model with values between 0 and 1. By an adjustable aggregation method, an overall quality value is derived.

MQC allows the user to implement a general quality model that complies with ISO 25010. The user can define quality properties, sub-characteristics and characteristics according to the project and process needs. Figure 6.4 shows this general hierarchy of the quality model.

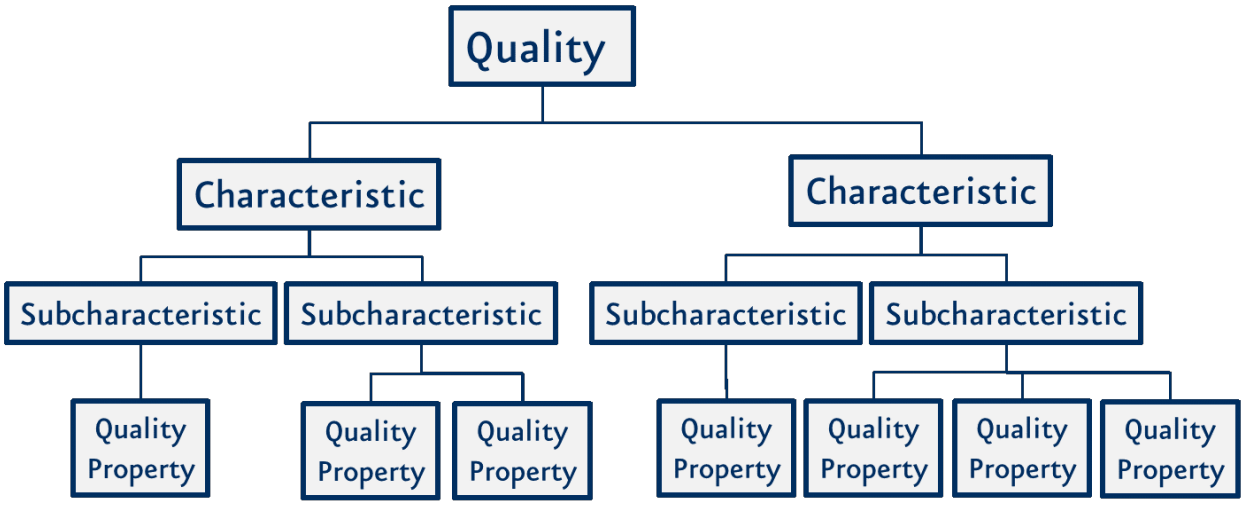


Figure 6.4: Quality Model – Concept (ISO 25010, p. 2)

In the following *Quality Aggregation* it is explained, how quality is aggregated on the different levels.

6.4 QUALITY AGGREGATION

In *Quality Computation*, we explained how the quality properties are computed for one artifact in a certain revision. Quality properties are represented by quality measures with a value between 0 and 1. Therefore, based on the imported data, quality values between 0% and 100% are computed.

Looking at the three dimensions 'Measure', 'Artifact' and 'Revision' all the quality measures build a quality cube, see Figure 6.5. Each quality property represents a slice (please note that several measurements yield several quality cubes).

Based on the quality measure for each quality property per revision and artifact, a first aggregation step is performed. For every single quality property, the corresponding quality measures for all artifacts are aggregated by calculating the average (see Figure 6.6).

This is the basis for further quality aggregation into sub-characteristics and characteristics.

According to the Quality Model defined by the user, quality property measures (previously aggregated over all artifacts) are aggregated to quality sub-characteristic. These once again are aggregated, this time to quality characteristics. A final step then combines all characteristics to a final quality value for each revision, see Figure 6.7. All aggregations use the average calculation as aggregation function, e.g. the average of all

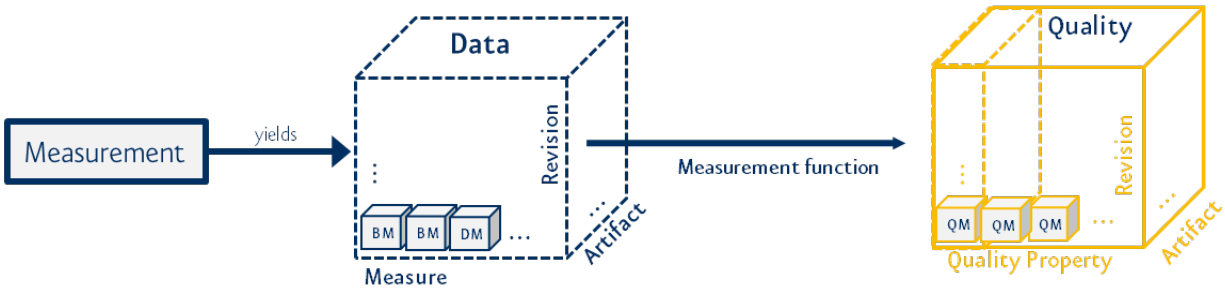


Figure 6.5: From data cube to quality cube

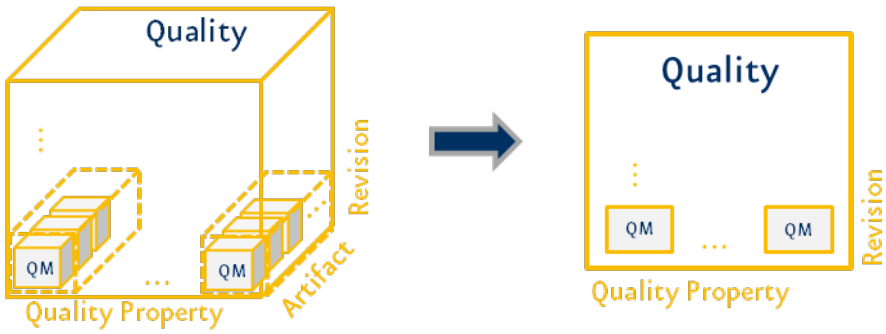


Figure 6.6: First aggregation step with respect to all artifacts

quality property measures for a certain sub-characteristic is calculated to gain the quality measure value for this particular sub-characteristic and so forth.

Note: How quality properties are structured affects the overall quality. Aggregation is performed with respect to the structure: The quality of a higher characteristic is based on the aggregation of the previous sub-characteristics or quality properties.

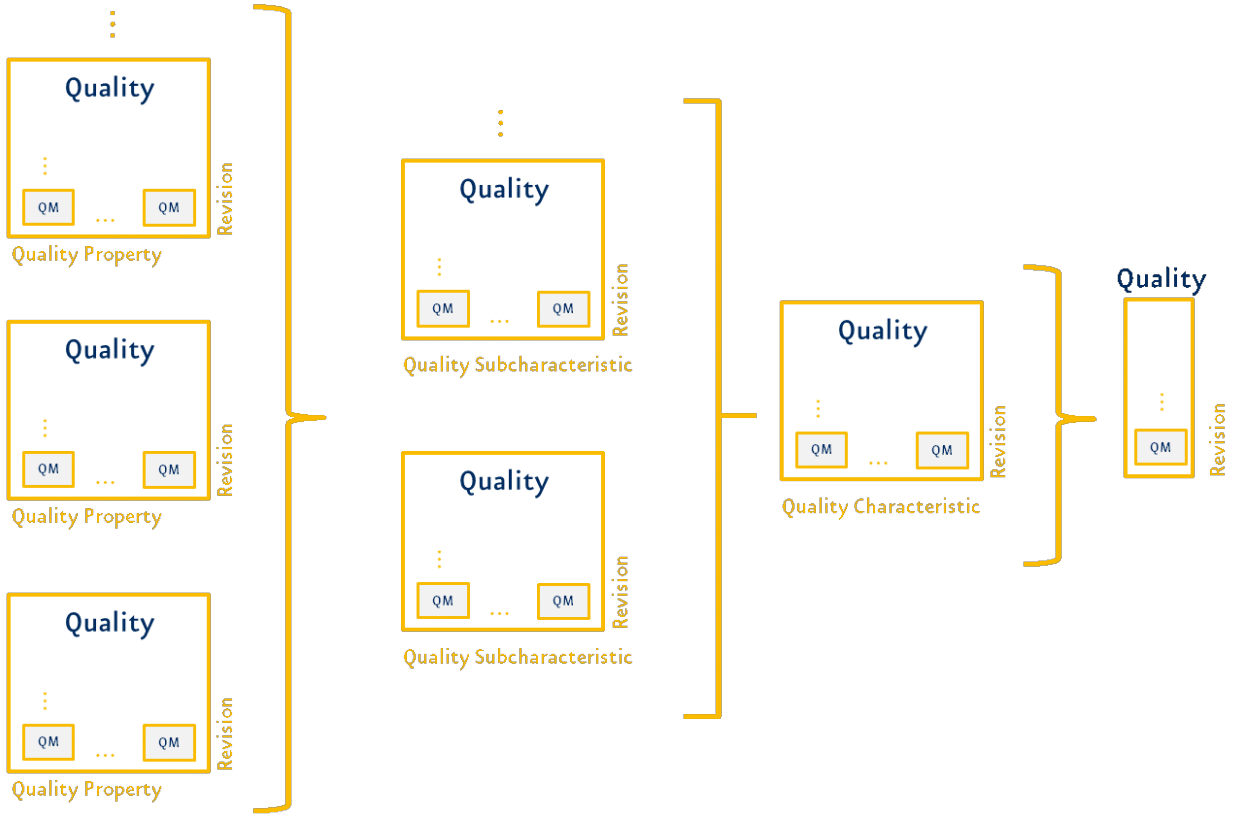


Figure 6.7: Quality aggregation that results in a final quality value for a revision

7 DATA SOURCES

This chapter gives you an overview about what Data Sources MQC supports by means of its adapters. Therefore, the user is able to keep track on those measures that are imported into MQC and can be used for the definition of the Quality Model.

The current release of MQC supports the following tools: *MES Model Examiner® (MXAM)*, *MES M-XRAY® (MXRAY)* and *MES Test Manager® (MTest)*, as well as *PikeTec TPT*, *Razorcat Tessy*, *MathWorks Polyspace* and *BTC EmbeddedTester*.

For the case there is no adapter, MQC puts at the user's disposal the possibility of a manual import of Base Measures (see *Manual data import*).

7.1 MES MODEL EXAMINER® (MXAM)

7.1.1 MXAM Data imported by MQC

MQC supports the standard MXAM report file (.mxmr).

The following example of an MXMR report describes which information is imported by MQC's MXAM adapter to MQC:

- from the `<RReport>` element, specifically the `date` element:
  - `ReportDateTime`
- from the `<artifacts>` element, the `<properties key="modelname">` element:
  - `ArtifactName`

MXAM provides guideline and finding result data for each artifact in the MXMR Report. Therefore, each artifact section will be parsed to get the information of the artifact result, model type and all findings and guideline results.

```
<artifacts result="Review" adapterId="com.modelengineers.mxam.tooladapter.matlab"␣
↪storageNature="Tool artifact">
<properties key="modelname" value="ReportReviewSldemo_wheelspeed_absbrake" visible=
↪"true"/>
<properties key="modeltype" value="MOD" visible="true"/>
<properties key="dynamic" value="" visible="true"/>
...
```

(continues on next page)

(continued from previous page)

```
  <summary itemType="Findings">
    <statistic resultType="Review" count="3"/>
    <statistic resultType="Failed" count="66"/>
    <statistic resultType="Info" count="28"/>
    <statistic resultType="Passed" count="12"/>
    <statistic resultType="Ignored" count="2"/>
  </summary>
  ...
  <summaries itemType="Guidelines">
    <statistic resultType="Review" count="1"/>
    <statistic resultType="Failed" count="5"/>
    <statistic resultType="Passed with Infos" count="24"/>
    <statistic resultType="Passed" count="11"/>
  </summaries>
</artifacts>
```

The adapter reads those measures to be found in the Findings and Guidelines header: `<Summary itemType="Findings">` and `<Summary itemType="Guidelines">`. Please, note that a MXAM report can contain various artifacts and for each artifact MQC reads out the Findings and Guidelines Summary, that are saved as `FindingCount` and `GuidelineCount`, respectively:

- `Review`
- `Failed`
- `Info` (for FindingCount) and `Passed with Infos` (for GuidelineCount)
- `Passed`
- `Ignored`
- `Aborted`
- `Canceled`
- `Repaired`
- `Unrepaired`
- `Warning` (for FindingCount) and `Warnings` (for GuidelineCount)

To get more insight of how these Base Measures are processed to a measurement function, please refer to *Quality Model for MXAM*.

7.2 MES M-XRAY® (MXRAY)

7.2.1 MXRAY Data imported by MQC

MQC supports the Standard XML MXRAY Report file.

The following information is extracted by the MQC/ MXRAY adapter:

- from the `<Timestamp>` element
  - `ReportDateTime`
- from the `<SubsystemQualityOverview>` header all included elements, usually these are:
  - `Local Complexity`
  - `Level`
  - `%Elementary Inputs Unused (globally)`
  - `Cyclomatic Complexity`
  - `Inports`
  - `Outports`

  For each of these Measures the variables `Good`, `Acceptable` and `Bad` are imported. Furthermore for `Local Complexity` the variables `LowerBoundOfAcceptable` and `LowerBoundOfBad` are read .
- from the `GlobalValueSummary` element
  - `Global Complexity (Ref0)`
  - `Global Complexity (Ref1)`
  - `Global Complexity (RefN)`
- from the `CloneGroups` element
  - `NumberOfDetectedCloneGroups`
  - `NumberOfSubsystemsAnalyzed`
  - `NumberOfUniqueSubsystemsInAllCloneGroups`
  - `NumberOfSubsystemsInAllCloneGroups`

Note: In MQC `CloneGroups.NumberOfUniqueSubsystemsInAllCloneGroups` is shown as `CloneGroups.Bad`:

`Bad` = `NumberOfUniqueSubsystemsInAllCloneGroups`

Additionally `CloneGroups.NumberOfSubsystemsAnalyzed` is used together with `CloneGroups.NumberOfUniqueSubsystemsInAllCloneGroups` to calculate a value for `CloneGroups.Good`:

`Good` = `NumberOfSubsystemsAnalyzed` - `NumberOfUniqueSubsystemsInAllCloneGroups`

To get more insight of how these Base Measures are processed to a measurement function, please refer to *Quality Model for M-XRAY*.

7.3 MES TEST MANAGER® (MTEST)

7.3.1 MTest Data imported by MQC

The MQC-MTest adapter supports the MQC-XML format for MTest Report files.

The MTest XML report consists of one `<DataEntryList>` header, which contains several `<DataEntry>` elements, each of them containing all the information for one Base Measure and Artifact.

The following information is extracted by the MQC/MTest adapter:

- from the `<RevisionDate>` element
  - `ReportDateTime`
- from the `<ArtifactNameOrAlias>` element
  - `ArtifactName`
- from the `<DataSourceNameOrAlias>` element the `BaseMeasureName` with its two variables (read out of the `<DataSourceValue>` element):
  - `Absolute` stored in MQC as `Reached`
  - `Reference` stored in MQC as `Total`

According to this pattern, from the `<DataEntryList>` element, the following Base Measures are imported with its respective variables (Reached and Total):

- `Assessment Work Progress`
- `Model Condition Coverage`
- `Model Decision Coverage`
- `Requirements Compliance`
- `Requirements with Reviewed Testability`
- `Reviewed Assessments`
- `Reviewed Test Sequences`
- `Test Sequence Work Progress`
- `Test Sequences Compliance`
- `Testable Requirements with Assessments`
- `Testable Requirements with Test Sequences`
- `Testable Requirements`

For the following Base Measures only exists one value that is stored in the variable `Reached`:

- `Total Assessments`
- `Total Requirements`
- `Total Test Sequences`

To get more insight of how these Base Measures are processed to a measurement function, please refer to *Quality Model for MTest*.

7.4 MANUAL DATA IMPORT

MQC provides the possibility to import data from any other data source using the manual import option.

The manual import has to be done in Excel. You can use this Excel template to create data that should be loaded to MQC. A template to import manual data in MQC, called `ManualDataImport_Template.xlsx` is provided within: `C:\Users\...\AppData\Local\TIBCO\Spotfire\10.3.3\Modules\MES.MQC.CoreExtension_X.X.X.X\Resources\SampleFiles\`.

Artifact	ReportDate	TPT.ModelTest.DecisionCoverageC1.Reached	TPT.ModelTest.DecisionCoverageC1.Total
EV3Control_de	13.08.2019	80	90
VehicleManage	13.08.2019	70	85

Figure 7.1: Sample file for manual import of two Base Measures for two Artifacts for the same revision

The column `Artifact` consists of entries representing the objects for which data shall be collected and for which quality shall be computed (e.g. Simulink models, requirements documents, software components).

The entries of the column `ReportDateTime` are considered by MQC as the days of raising the data, which shall be collected.

Using the syntax `DataSource . Measurement . Measure . Variable`, you are able to load into your project any data you want. Use `DataSource` to specify where your data is coming from (e.g. the name of the tool which produced the data). You can optionally fill out `Measurement` to provide more structuring regarding the data (e.g. the reason why data is collected, the test environment or contexts models have been through, etc). Use the `Measure` to provide more structuring of your data regarding the result of the measurement. To do that, give a name (the entry of the Measure field) and a `Variable`. Finally, the `Value` contains the metric value that you would like to import for the artifact and the related revision. It is recommended to fill out `Measurement` to achieve a high compliance to ISO 250xx and to avoid that a default name is provided by MQC, which has probably no meaning to you.

As shown in Figure 7.1 you should put your variables using `DataSource . Measurement . Measure . Variable` syntax within the excel column name. The value for the respective `Artifact` for a `ReportDateTime` has to be assigned to the corresponding line.

In case you might want to assign default values for your imported Base Measures, you have to define them in compliance with this syntax within the Quality Model Base Measure excel sheet (see Figure 11.1).

7.5 PIKETEC TPT

TPT is considered an (xml-based) external test tool for which an MQC adapter executes the following operations presented by the example of the following extract of an TPT example XML report:

```
<Header ExecutionConfig="Lights Control MATLAB" ExecutionDate="14:47:58 10.05.2016"␣
↪TptFileName="D:\requirements.tpt" TptVersion="8u2">
  <Property Name="Model Under Test" Value="D:\matlab-platform\lights_control_simulink.
↪mdl"/>
  <Property Name="System Under Test" Value="lights_control_simulink/lights_control"/>
  <Platform History="100" Name="MATLAB-Platform" Stepsize="10000" Timeout="60000000">
    <Property Name="MATLAB Version" Value="MATLAB 8.4"/>
  </Platform>
</Header>
...
<Summary AssessmentDuration="2.518" ExecutionDuration="2.078">
  <ExecutionSummary Errors="0" Failed="6" Inconclusive="0" Succeeded="5" Tests="11"/>
</Summary>
```

MQC reads out

- from the main `<Header.. >`
  - `ExecutionDateTime` (stored in MQC as `ReportDateTime`)
  - `SystemUnderTest Value` (stored in MQC as `ArtifactName`), in this case "lights_control_simulink/lights_control". Please, note that TPT stores in the XML only the name (instead of the complete path) of the subsystem
  - `TptVersion`
  - `PlatFormName`
- from the `<ExecutionSummary>` header the categories
  - `Tests` (stored in MQC as `TestCount.Test`)
  - `Succeeded` (stored in MQC as `TestCount.Succeeded`)
  - `Failed` (stored in MQC as `TestCount.Failed`)
  - `Errors` (stored in MQC as `TestCount.Errors`)
  - `Inconclusive` (stored in MQC as `TestCount.Inconclusive`)

MQC offers an additional structuring method, called `Measurement` described in Section 7.4. Herewith, different Measurements with the same BaseMeasure and Variable name can be read out. You can assign the measurement name of MiL, SiL or PiL, which can be stated within the file name or the `PlatFormName` with the following syntax `_SIL_`.

7.6 RAZORCAT TESSY

From the extract of the Tessy Example XML report,

```
<report success="notok" tessy_version="4.0.15" xml_version="3">
<statistic notexecuted="0" notok="11" ok="54" total="65">
  <category count="54" name="ok"/>
```

(continues on next page)

(continued from previous page)

```
  <category count="11" name="notok"/>
  <category count="0" name="notexecuted"/>
</statistic>
<info date="2018-08-23" time="16:20:30+0200"/>
...
<tessyobject id="1024" level="0" name="Testsuite" success="notok" type="project">
```

MQC reads

- from the main (`report`) header:
  - `tessy_version` (stored in MQC as `TessyReportVersion`)
- from the `<statistics..>` header the categories:
  - `ok`
  - `notok`
  - `notexecuted`.
- from the `<info..>` header:
  - `date` and `time` (stored in MQC as `ReportDateTime`)
- from the `<tessyobject..>` header:
  - `name` (stored in MQC as `ArtifactName`)
  - `type` (stored in MQC as `TessyObjectType`)

The Tessy data is imported to the transformed ValueFact table maintaining the same notation of the Tessy classification of `ok`, `notok` and `notexecuted`, yet assigning them to the BaseMeasure `TestCount`.

7.7 MATHWORKS POLYSPACE

7.7.1 Polyspace Xml-adapter

If a Polyspace Xml report is created, MQC reads out from the xml-file:

- `PubDateTime` (stored in MQC as `ReportDateTime`)
- `Subtitle` (stored in MQC as `ArtifactName`)
- `PolyspaceVersion`
- `MISRA-C Checker.Violations`
- `MISRA-C Checker.Pass_Fail`

This xml-file refers to several xfrag-files in the `Polyspace-doc` directory to be found on the same level as the xml-file.

From the `image-000-chapter.xfrag`-file,

```
<title>Run-Time Checks Summary</title>
  <tgroup align="left" cols="2"><colspec colname="1" colnum="1" colwidth="600*"/>
↪<colspec colname="2" colnum="2" colwidth="600*"/><thead>
  <row bgcolor="#6A80BF">
    <entry>Run-Time Checks</entry><entry>Enabled</entry><?dbfo bgcolor="#6A80BF"?></
↪row></thead>
  <tbody>
  <row bgcolor="#F0F2F9">
    <entry>Number of Red Checks</entry><entry>
    <emphasis role="red">0</emphasis></entry>
    <?dbfo bgcolor="#F0F2F9"?>
  </row>
  <row bgcolor="#FAFBFE">
    <entry>Number of Gray Checks</entry>
    <entry><emphasis role="gray">0</emphasis></entry><?dbfo bgcolor="#FAFBFE"?>
  </row>
  <row bgcolor="#F0F2F9">
    <entry>Number of Orange Checks</entry>
    <entry><emphasis role="orange">0</emphasis></entry>
    <?dbfo bgcolor="#F0F2F9"?></row><row bgcolor="#FAFBFE">
    <entry>Number of Green Checks</entry>
    <entry><emphasis role="green">220</emphasis></entry><?dbfo bgcolor="#FAFBFE"?>
  </row>
  <row bgcolor="#F0F2F9">
    <entry>Proven</entry>
    <entry>100.0%</entry><?dbfo bgcolor="#F0F2F9"?>
  </row>
  <row bgcolor="#FAFBFE"><entry>Pass/Fail</entry>
  <entry>-</entry><?dbfo bgcolor="#FAFBFE"?></row></tbody>
  </tgroup>
</table>
```

MQC extracts the following information, modifies and transformes it via predefined data table transformations:

- `Number of Red Checks` (stored as `Run-Time Checks.Major` in MQC)
- `Number of Gray Checks` (stored as `Run-Time Checks.Minor` in MQC)
- `Number of Orange Checks` (stored as `Run-Time Checks.Moderate` in MQC)
- `Number of Green Checks` (stored as `Run-Time Checks.Good` in MQC)
- `Percentage of Proven` (stored as `Run-Time Checks.Proven` in MQC)
- `Pass/Fail` (stored as `Run-Time Checks.Pass_Fail` in MQC)

7.7.2 Polyspace Text-adapter

If a Polyspace tab-separated Text report is created, MQC reads from this txt-file:

- `Last modified time of file` (stored in MQC as `ReprtDateTime`)
- `File`
- `Family`
- `Color`

From the `File` column MQC extracts the file paths of all files used to create the report. MQC then obtains the common prefix from these file paths and extracts the last directory from the prefix.This is stored as the `Artifact Name` in MQC. For example if paths read from the "File" column are as follows:

- `E:\D42\A987R\InputOut\Swc_HMK`\TLSim\Rte.c
- `E:\D42\A987R\InputOut\Swc_HMK`\TLProj\TL_Swc_WIM\Swc_WIM.c
- `E:\D42\A987R\InputOut\Swc_HMK`\TLSim\J_GLOBAL_STD.h

The common prefix is marked above and MQC extracts the last common directory `Swc_WIM` as the artifact name.

MQC extracts from the tab-separated columns `Family` as base measure and aggregated `Color` count as variable. MQC then modifies and transforms it via predefind data table transformations as follows:

- `Family:Run-time Check and Color:Red` (stored as `Run-Time Checks.Major` in MQC)
- `Family:Run-Time Check and Color:Gray` (stored as `Run-Time Checks.Minor` in MQC)
- `Famliy:Run-Time Check and Color:Orange` (stored as `Run-Time Checks.Moderate` in MQC)
- `Family:Run-Time Check and Color:Green` (stored as `Run-Time Checks.Good` in MQC)
- `Family:Global Variable and Color:Gray` (stored as `Global Variable.Unused variable` in MQC)
- `Family:Global Variable and Color:Not Applicable` (stored as `Global Variable. Used non-shred variable` in MQC)
- `Family:MISRA-C` (stored as `MISRA-C Checker.Violations` in MQC)

7.8 BTC EMBEDDEDTESTER

MQC supports two types of Embedded Tester (ET) report formats:

- *Standard XML report format*
- *MQC XML report format* (previous versions of ET)

7.8.1 Standard XML report format

If a standard XML report is created, MQC will read from the XML file:

```
<BTCXmlReport>
  <ProfileInfo lastArchitectureUpdate="Wed Nov 07 16:32:55 CET 2018"␣
↪lastB2BTestExecution="Sat Aug 11 08:44:00 CEST 2018" lastRBTSILTestExecution="Wed␣
↪Nov 07 16:39:43 CET 2018" lasteModifier="mes" modelName="TestObject.slx"␣
↪modelVersion="0.0.1+STD" profileName="UT_TestObject.epp"/>
  <RBT>
    <Tests sumSILErrorExecutions="0" sumSILFailedExecutions="0"␣
↪sumSILMissingExecutions="0" sumSILOutdatedExecutions="0" sumSILPassedExecutions="7"␣
↪sumTestCases="7"/>
    <Requirements percentageSILPassedRequirements="100.0" sumRequirements="11"␣
↪sumSILFailedRequirements="0" sumSILMissingOrOutdatedStatusRequirements="0"␣
↪sumSILPassedRequirements="11"/>
    <SILFailedRequirements/>
    <RBTCoverageOverview>
      <Statement covered="77.33" handled="78.95" unknown="21.05" unreachable="1.62"/>
      <Decision covered="62.82" handled="65.38" unknown="34.62" unreachable="2.56"/>
      <MCDC covered="66.28" handled="68.6" unknown="31.4" unreachable="2.33"/>
    </RBTCoverageOverview>
  </RBT>
  <B2B>
    <Tests lastB2BTestName="TL MIL vs SIL" status="FAILED_ACCEPTED" sumErrorTests="0"␣
↪sumFailedAcceptedTests="32" sumFailedTests="0" sumPassedTests="25" sumTests="57"/>
    <B2BCoverageOverview>
      <Statement covered="98.38" handled="100.0" unknown="0.0" unreachable="1.62"/>
      <Decision covered="97.44" handled="100.0" unknown="0.0" unreachable="2.56"/>
      <MCDC covered="97.67" handled="100.0" unknown="0.0" unreachable="2.33"/>
    </B2BCoverageOverview>
  </B2B>
</BTCXmlReport>
```

MQC extracts the following information, stores and transformes it to the MQC data structure, so that the imported data can be added to the ValueFact table as rows:

- from the `<ProfileInfo>` header:
  - `lastRBTSILTestExecution` (stored in MQC as `ReportDateTime`)
  - `modelName` (stored in MQC as `ArtifactName`)
- from the `<RBTCoverageOverview>` header:
  - `MCDC covered` (stored in MQC as `Requirement Based Testing.MCDC`)
- from the `<RBT>` header:
  - `percentageSILPassedRequirements` (stored in MQC as `Requirement Based Testing.Test coverage`)
- from the `<BTB>` header:
  - `sumTests` (stored in MQC as `Structural Testing.Nb Total`)
  - `sumPassedTests` (stored in MQC as `Structural Testing.Passed`)

  - `sumFailedAcceptedTests` (stored in MQC as `Structural Testing.Failed Accepted`)
  - `sumFailedTests` (stored in MQC as `Structural Testing.Failed`)
  - `sumErrorTests` (stored in MQC as `Structural Testing.Error`)
  - `MCDC covered` (stored in MQC as `Structural Testing.MCDC`)

7.8.2 MQC XML report format

If the MQC XML report is created, MQC is able to automatically import a great number of Base Measures:

- `Back-2-Back-Results` with the following variables: - `Errors` - `Failed` - `Failed (Accepted)` - `Passed` - `Total Vectors`
- `Code Coverage - Condition Coverage` with the following variables: - `Covered` - `Handled` - `Tests` - `Unreachable (n/inf)`
- `Code Coverage - Decision/Branch Coverage` with the following variables:
  - `Covered`
  - `Handled`
  - `Tests`
  - `Unreachable (n/inf)`
- `Code Coverage - Modified Condition/Decision Coverage` with the following variables:
  - `Covered`
  - `Handled`
  - `Tests`
  - `Unreachable (n/inf)`
- `Code Coverage - Statement Coverage` with the following variables:
  - `Covered`
  - `Handled`
  - `Tests`
  - `Unreachable (n/inf)`
- `Requirements Coverage` with the following variables:
  - `Requirements`
  - `Requirements Covered`
  - `Requirements Fulfillment`
- `Test Execution Results` with the following variables:
  - `Errors`

- Failed
- Failed (Accepted)
- Passed
- Total Vectors

8 ACTIONS AND CONFIGURATION MENU

Per default the menu is enabled on all MQC pages. You can simply switch it on or off by clicking the `Menu` icon within the MES Quality Commander banner in the top-left.

Figure 8.1: Enable the menu within the MES Quality Commander banner

The MQC Menu is divided into Actions and the Configuration.

8.1 ACTIONS

8.1.1 Refresh Data

Once you have configured *Data Sources* and imported data, you can gradually import your file reports by copying new files / directories into the configured directories. To consider your newly added files/directories, press the "Refresh" Button.

Alternatively, you can enable the Automatic Refresh checkbox, so that MQC refreshes the specified directories and files automatically when changes are detected.

MQC searches all configured files and subdirectories recursively to load new file reports into the collection of your analysis.

Changes inside the added directories (new files, removal of files or even changes within the files) are automatically detected.

When clicking on the Refresh button the message will show to the "Data is up-to-date" status message, after the data has been refreshed.

8.1.2 Create Report

In this dialog you can create a Status Report (see *Status Report*).

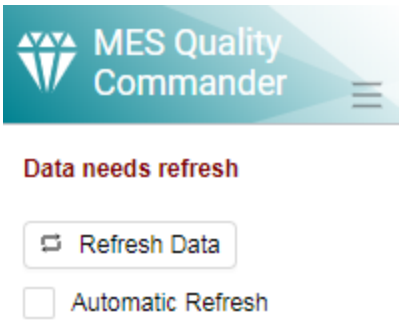


Figure 8.2: The "Needs refresh" status message appears due to changes within the monitored directories.

8.1.3 Data of the Quality / Filtered by Quality

In this dialog you get a drill down from Quality to Data (see *Data Details Drill Down*).

8.2 CONFIGURATION

The bottom of menu contains all configurations of the analysis.

8.2.1 Data Sources

Users can import either single files or directories by clicking on the *Data Sources* section and select files or directories that contain file reports, e.g. our example data that was delivered to you in section *Quick Start Guide*.

8.2.2 Project structures

Users can load, export or reload the Project Structures excel configuration (please refer to *Project Structures Configuration* for more information).

8.2.3 Quality Model

Users can load, export or reload the Quality Model excel configuration. A default Quality Model is provided with MQC (please refer to *Quality Model Configuration* for more information).

8.2.4 Target Values

Users can load, reload or remove the Target Values excel configurations (please refer to *Target Values* for more information).

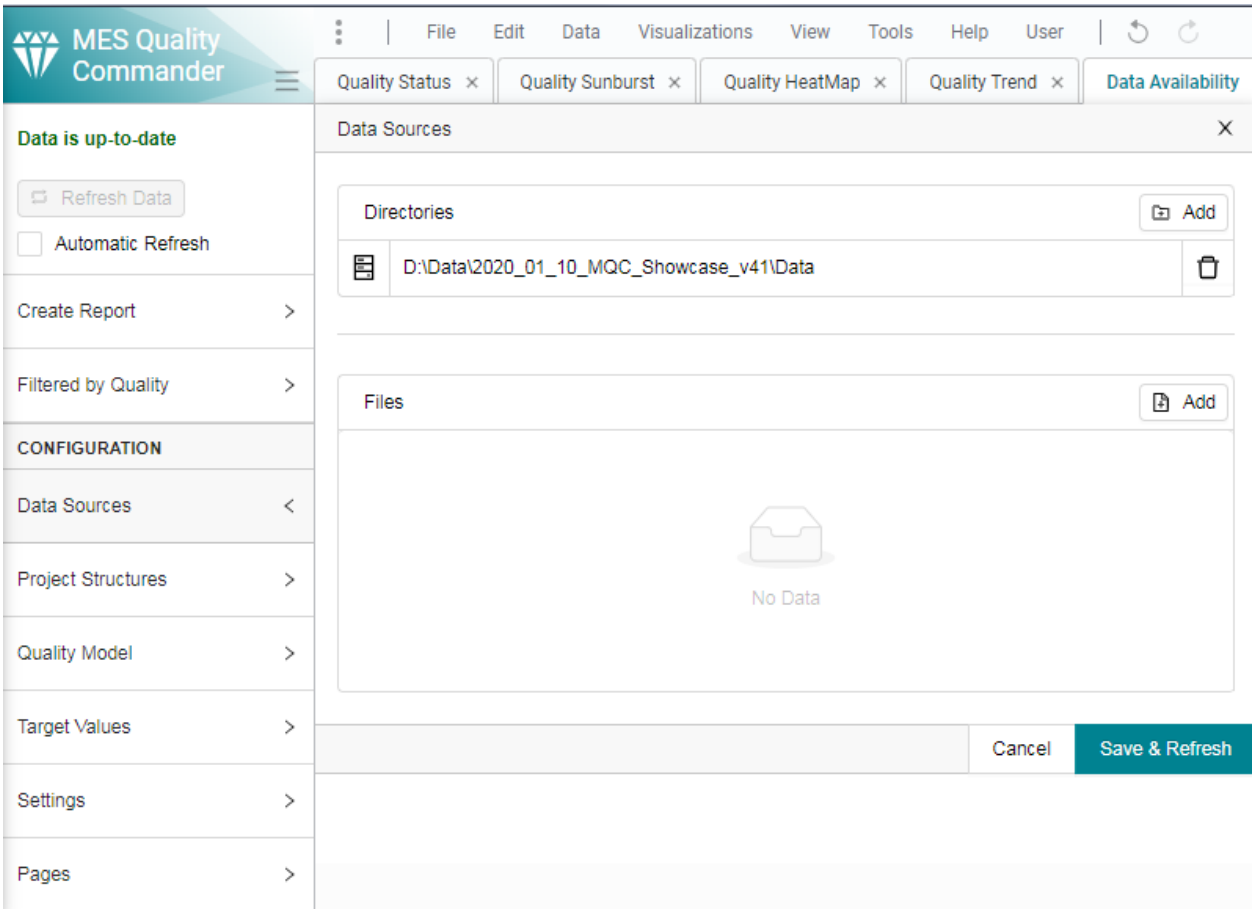


Figure 8.3: Configure Data Sources by adding files and/or directories

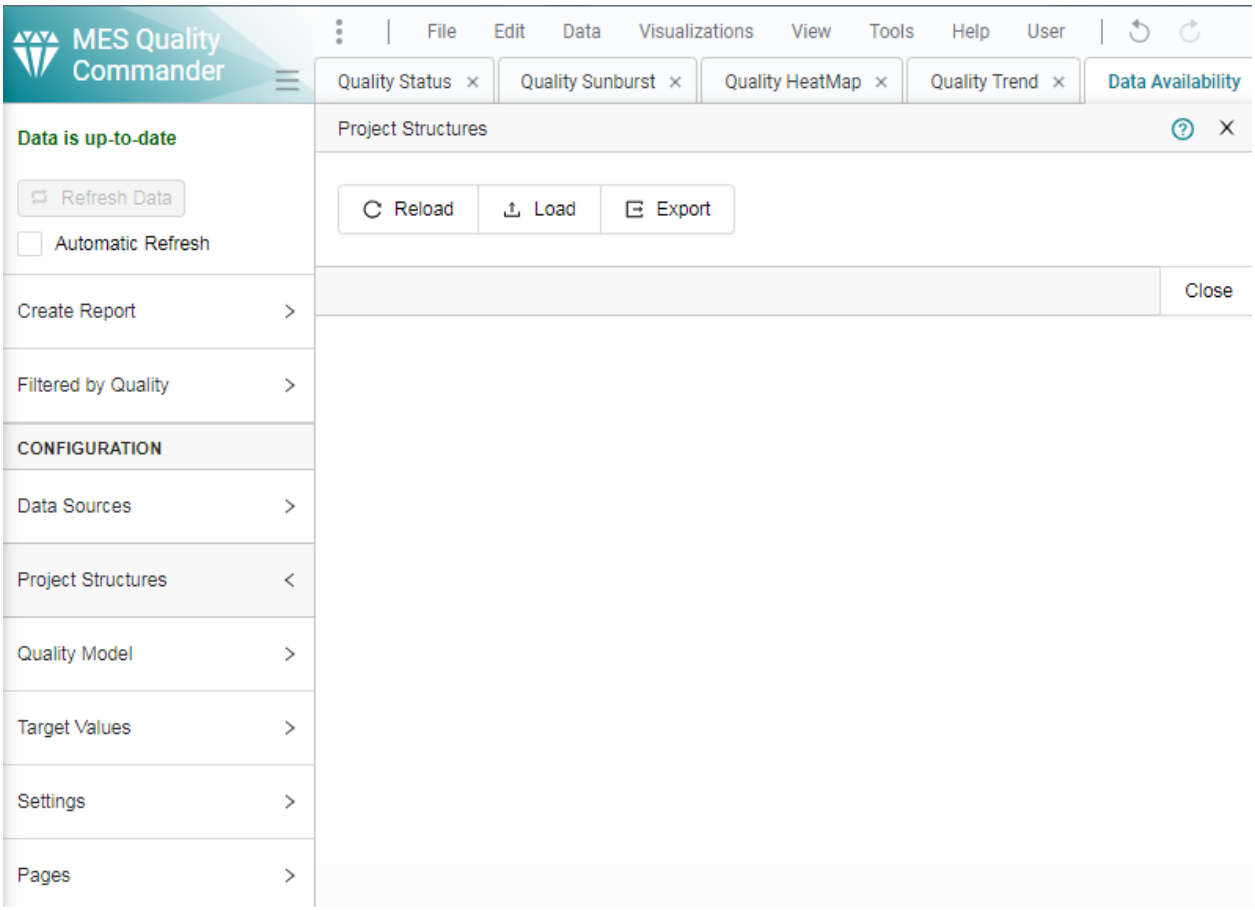


Figure 8.4: Configure Project Structures

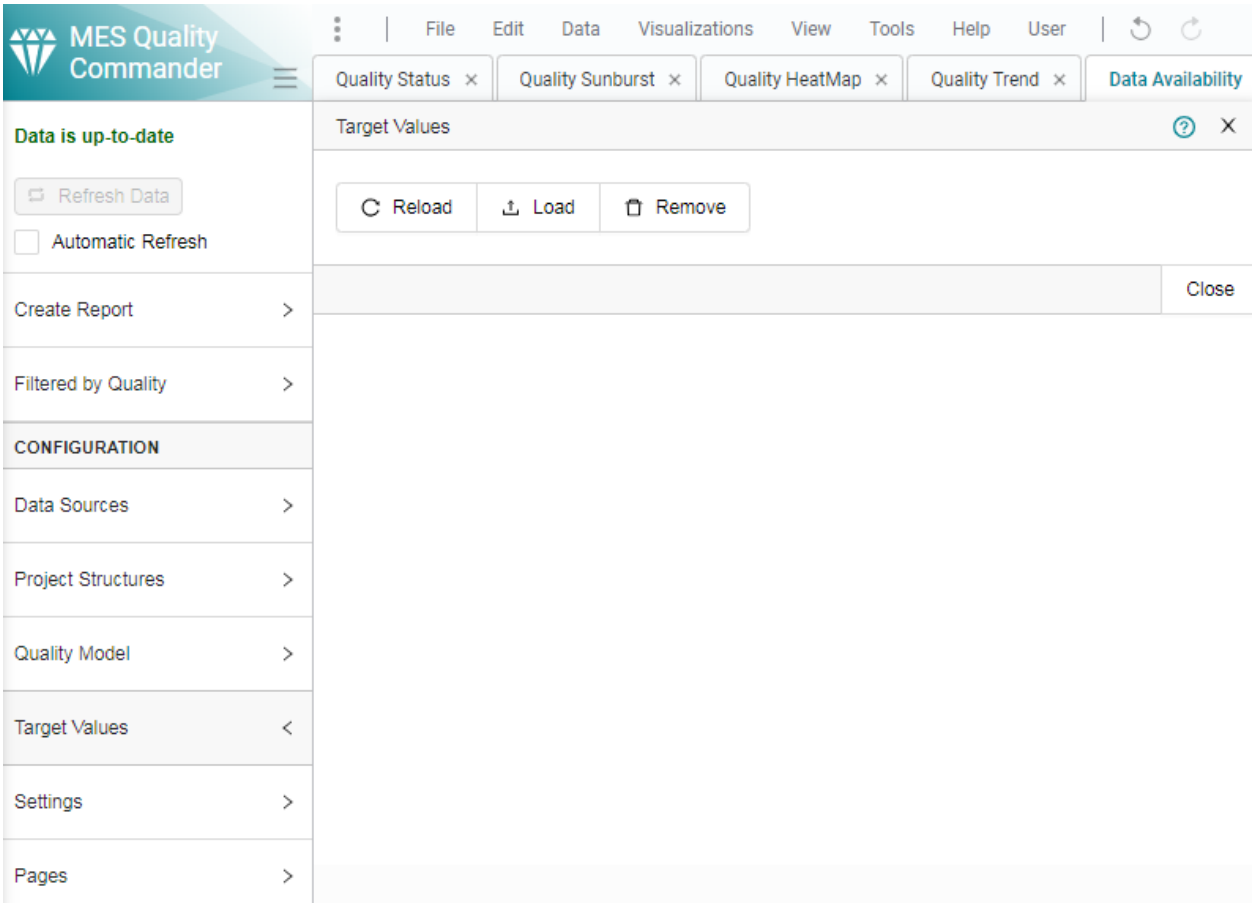


Figure 8.5: Configure Target values

Note: In the *Project Structure*, *Quality Model* and *Target Values* dialogs the *Reload* button is only visible when a configuration file has already been imported. A tooltip shows the paths to the currently loaded file.

8.2.5 Settings

This section contains a group of settings:

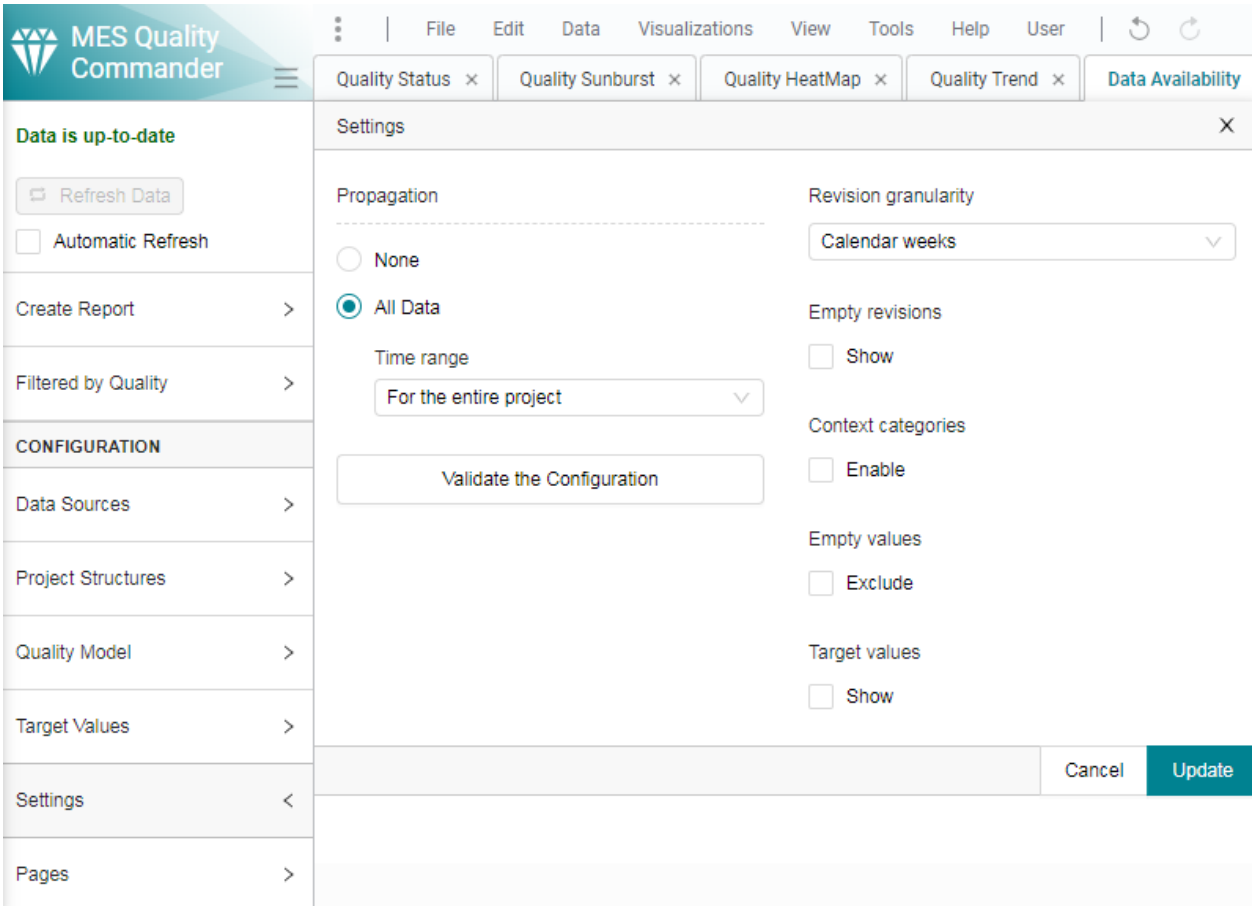


Figure 8.6: Settings dialog

Propagation

(see *Data Propagation*)

Revision granularity

Can be used to change the degree of compactions of the revisions for which data exists to better visualize your trend line charts. The revision granularity can be selected depending on your needs (see Figure 8.7).

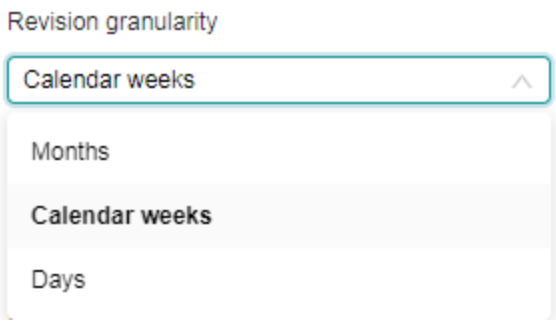


Figure 8.7: The default revision granularity is CalendarWeeks, but it might be useful to change it to *Days* or *Months*

Show Empty Revisions

(see *Project Milestone Structure*)

Enable Context Categories

(see *Define your Context Categories*)

Exclude Empty Values

(see *Quality Properties*)

Show Target Values

(see *Configuration of Target Values*)

Validate the configuration

(see *Validation of Configuration*)

8.2.6 Pages

This section contains all pages that exist in MQC, categorized in four groups (Quality, Data, Tool and Custom pages). You can add or remove them by selecting or deselecting the checkboxes.

- Quality pages include Quality Status, Quality Sunburst, Quality HeatMap, Quality Trend, Quality Model pages.
- Data pages include Data Availability, Data Status, Data Trend pages.
- Tool pages include MXAM, MTest, MXRAY, Polyspace, which contains data from these data sources (see *Tool Pages*).

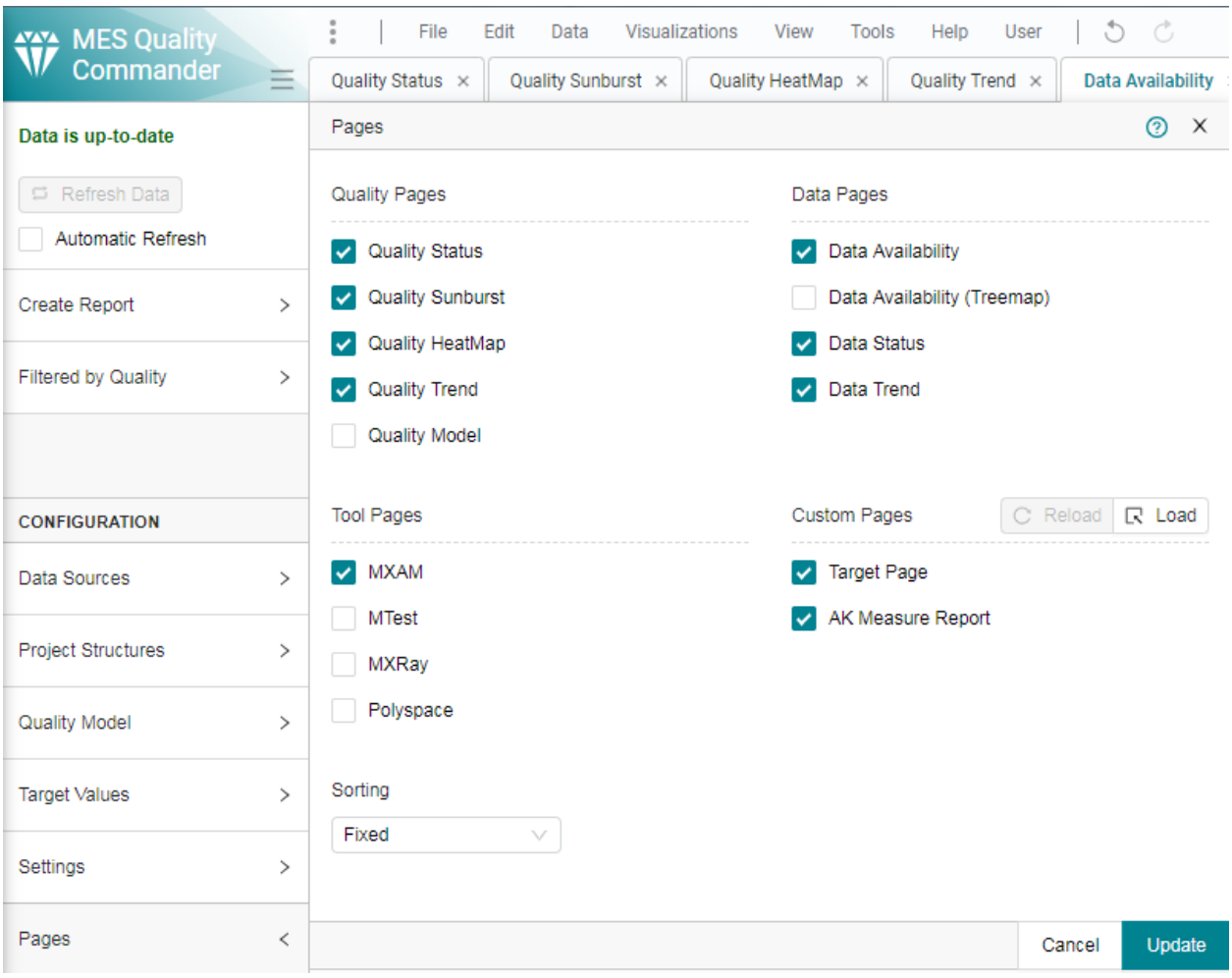


Figure 8.8: Pages dialog

- Custom pages: are defined in *Custom Pages*.
- Sorting: there are two options for sorting pages (see *Page order in the Pages dialog*). By default the pages are in *Fixed* order. If *Custom* is selected for ordering, new pages are added after existing pages.

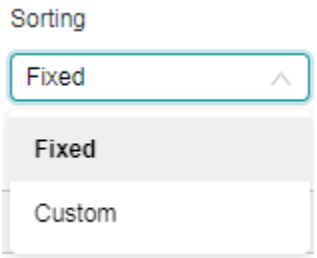


Figure 8.9: Page order in the Pages dialog

Note: To apply changes in the Settings and Pages dialogs you have to click on the `Update` button.

9 PAGES AND VISUALIZATIONS IN DETAIL

In this chapter, we will take you through the steps involved in performing a detailed quality evaluation in MQC. This will give you a good impression of the functional details of MQC. At the same time, it helps you get accustomed to the tool in a more detailed way than in *Quick Start Guide* where a quick introduction to the basic functionality of MQC is given.

Building on this and taking into account the dashboards for a project conclusions about the development of software projects are described. Based on these conclusions MQC shows possible reasons for bad quality and helps you deducing actions to carry out for improving quality.

9.1 THE MQC DASHBOARD

MQC consists of a default dashboard of overview pages that contain all the important information of your project. The dashboard consists of data and quality pages, showing trend and status respectively, i.e. *Quality Trend* and *Quality Status*, as well as *Data Trend* and *Data Status*.

The *Quality Trend* page will give an overview of the quality of your project, whereas the *Quality Status*, *Quality Sunburst* and *Quality Heatmap* page provides an in-depth analysis of Artifacts, i.e. an overview of the quality of each Artifact.

The *Quality Model* page gives information about the definition of your Quality Properties. It will show you any kind of validation errors.

The *Data Status* page will give you an overview of the data of your project. By means of Data Details functionality (*Data Details*) you will get from quality pages, such as *Quality Status* or *Quality Sunburst* to the data pages to track the source of a measurement, i.e. going one level down on data source level to *Data Trend* where you can see your raw data's trend and get an overview of your related base measures.

The general concept of these pages is that all the visualizations are interactive: If you click into any of the visualizations, the other visualizations will react in a top-down / left-right manner and then show information related to what you have marked. You can do the major selections on the left and top sections of the visualization window and the effect can be seen in the bottom right visualizations.

Marking is cumulative. The marked elements of the first chart will not reset if you mark something in the next chart later. The effect is a combination of marking done in both the charts. To go back to normal state click on the chart where there are no elements.

In dependence of the Project Structures you have imported, the right side panel will display options for filtering the data.

In the following subsections the four main pages as mentioned above are described:

9.2 DATA PAGES

9.2.1 Data Trend

By means of the *Data Trend* page, you can track the source of a measurement, going one level down compared to the *Data Status* page, i.e. on data source level. You can see your raw data's progression and processing getting an overview of your derived and related base measures.

It consists of the following selecting windows:

- *Projects* (on the top left): Select your project as defined in your project structure. Notice, that you see an availability trend inside the tile. A gradient availability coloring is used that depends on the availability of all measure values for all artifacts shown for the most recent revision.
- *Artifacts* (Bottom left): Select the artifacts of your choice. By default, all are selected. A gradient availability coloring is used that depends on the availability of the measure values for a single artifact for the most recent revision.
- *Measures* (next to Artifacts): The count of the measures is with respect to all artifacts and for the last revision. It uses a gradient availability coloring.
- *Availability Distribution*: This is a bar chart, which bins the amount of available, missing and propagated data for each revision... A categorical availability coloring is used.

The *Measure Trend* visualization shows per Artifact for each measure value (base measures and derived measures) a trend over all revisions. If a certain revision is selected, the data value for this specific revision is shown only. It is possible to only show the trends for a single or for selected artifacts as well as for selected measures. The Measure Trend visualization uses a dedicated color per trend line for each measure value.

9.2.2 Data Status

The *Data Status* page offers the following selecting windows:

- *Projects* (on the top left): Select your project as defined in your project structure. Notice, that you see an availability trend inside the tile. A gradient availability coloring is used that depends on the availability of all measure values for all artifacts shown for the most recent revision.
- *Artifacts* (Bottom left): Select the artifacts of your choice. By default, all are selected. A gradient availability coloring is used that depends on the availability of the measure values for a single artifact for the most recent revision.
- *Availability Distribution*: This is a bar chart, which bins the amount of available, propagated, missing data and data to which a default value is applied for each revision. It uses a categorical availability coloring.

The Availability Treemap is the reacting window to the selections (intersections) made by the marking in the selecting windows mentioned above. It also uses a categorical availability coloring.

9.3 QUALITY PAGES

9.3.1 Quality Status

The *Quality Status* page provides an in-depth analysis of Artifacts. Therefore, it will give an overview of the quality of each Artifact.

Selecting windows are:

- *Projects* (on the top left): Select your project as defined in your project structure for which you want to see quality. If you have defined only a single project, it is selected by default. You can reset the marking for the entire page at any point by clicking on this tile. Notice, that you see a quality trend over all revisions inside the tile. A gradient quality coloring is used that depends on the overall quality for all artifacts, shown for the most recent revision.
- *Artifacts* (Bottom left): Select the Artifacts of your choice. By default, all are selected. The values within the tiles are always related to the latest revision. A gradient quality coloring is used that depends on the quality for each artifact, shown for the most recent revision.
- *Quality properties* (next to Artifacts): Select the Quality Property. Note, that the Properties tiles are listed in ascending order of quality measure. If the quality measure cannot be computed, then it appears in the bottom of the list with an empty value. The values within the tiles are always of the latest revision. A gradient quality coloring is used.
- *Quality Bin Distribution* (top): The computed quality property measures are mainly binned as Good, Acceptable and Bad. By marking the bins in this visualization you can do selections based on the computed quality status of all quality properties per revision. You can also select the revision of the data by selecting all the bins of the same revision. For example, if you select the green bin of the latest revision on the Quality Bin Distribution visualization, the reacting window shows an overview of all information related to good quality measures for this revision. For the bins a categorical quality coloring is used.

The reacting window is a heatmap visualization plotted for each Artifact against each Quality Property. Each Quality Property tile has a gradient quality coloring, which represents the quality value for the particular property. Based on the marking from the selecting windows, the boxes in the heatmap are highlighted.

You can now start identifying the reason for bad quality. The general concept of identifying issues is to click on (or hover over) red quality bins or tiles. When clicking on red, data that corresponds to green and yellow bins is excluded from the chart visualizations to receive a first impression of the reason for bad quality, concretely the quality properties that have failed.

9.3.2 Quality Sunburst

The *Quality Sunburst* page will give an overview of the quality of your project.

The selecting windows are the same as on the *Quality Status* page, just the main visualization shows the sunburst visualization where the outer ring consists of all quality properties defined by the quality model (see section *Quality Properties*).

9.3.3 Quality Heatmap

The *Quality Heatmap* page will give an overview of the quality of your project.

The selecting windows are the same as on the *Quality Status* page, just the main visualization shows a more detailed heat map visualization, where quality properties and artifacts are visualized with their aggregation and respective "size" (weight).

9.3.4 Quality Trend

The *Quality Trend* page will give an overview of the quality of your project.

The selecting windows are the same as on the *Quality Status* and *Quality Sunburst* page.

Quality Trend by Artifact and Quality Trend by Property are the reacting windows. These line chart visualizations are updated based on your previous selections.

The trend visualizations use a categorical coloring, so each line representing an Artifact resp. a quality property uses a dedicated color.

9.4 TOOL PAGES

9.4.1 MXAM Tool Page

Besides the standard data and quality pages, MQC provides an additional tool page showing details on data provided by MXAM.

From the menu bar choose **Tools/MQC Tool Pages/Add MXAM Details Page** to add the MXAM tool page.

The main visualization (horizontal bar chart) shows for each artifact all findings based on FindingCount (e.g. `Failed`, `Passed`, `Warning` etc. as provided by the MXAM report). As long as data for multiple revisions is imported and no particular revision is marked respectively, the visualization offers to scroll between revisions to get the finding status for a certain point in time.

The Artifact KPI on the left-hand side shows a tile for each artifact colored according to the worst finding result for this artifact (using the MXAM results order), i.e. in Figure 9.1 all artifacts have failed checks, but no findings for `Canceled`, `Aborted` or `Review`. So, all tiles are colored red.

The MXAM result coloring is shown in the following figure:

Additionally each tile shows per artifact:

- the number of severe issues (sum of `Aborted`, `Canceled`, `Failed`, `Review` and `Unrepaired` findings)
- the number of issues (servere issues plus `Warning`)

The Project KPI chart on the top-left of the page shows the color of the worst finding over all artifacts as well as the number of severe issues and issues summed up over all artifacts.

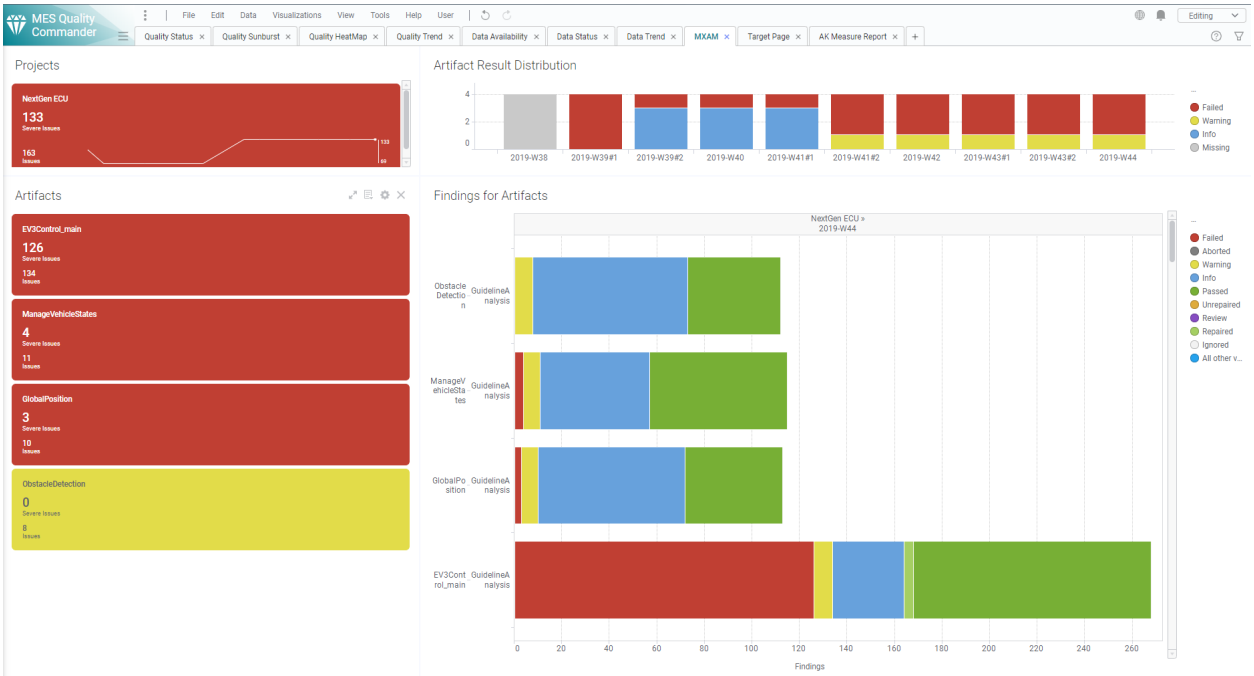


Figure 9.1: MXAM Details Page showing Findings per Artifact and Revision

Canceled	The analyses was canceled by the user and the check was not executed.
Aborted	The check was not executed because of a wrong configuration or runtime error.
Review	The element cannot be automatically checked. The user must manually review the element.
Failed	A faulty element was detected which violates the guideline.
Unrepaired	The automatic repair of the element failed.
Warning	The guideline violation of the element is not considered as critical.
Info	The check returns an information for the user for documentation purposes.
Repaired	The element was repaired successfully .
Passed	The element passed the check and is compliant to the guideline.
Ignored	The element was ignored by the check because it is on the ignore-list.

Figure 9.2: List of all existing MXAM results

The distribution chart (top-right) shows for each revision the number of artifacts binned according to their worst findings. For the example in Figure 9.1 this means there are four artifacts in total, all of them with `Failed` findings.

By these means the user gets an overview on

- how many findings per artifact exist
- which findings per artifact exist
- what are the worst findings to concentrate on first (for each artifact and for the whole project)
- and how this has evolved over time (e.g. from project start time until actual date).

9.4.2 MXRAY Tool Page

Besides the standard data and quality pages, MQC provides an additional tool page showing details on data provided by MXRAY.

From the menu bar choose **Tools/MQC Tool Pages/Add MXRAY Details Page** to add the MXRAY tool page.

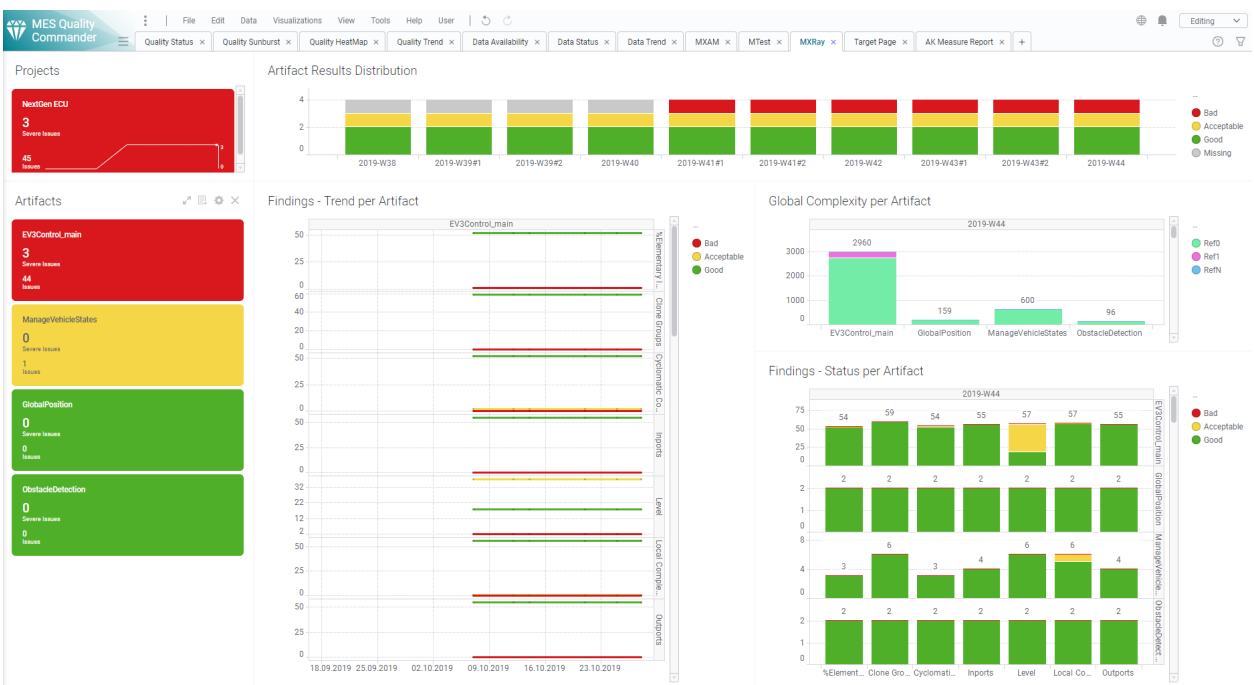


Figure 9.3: MXRAY Details Page showing Findings in Trend and Status and Global Complexity per Artifact and Revision

This page contains three main visualizations:

- Global Complexity per Artifact
- Status of subsystem results per base measure and Artifact
- Trend of subsystem results per base measure and Artifact

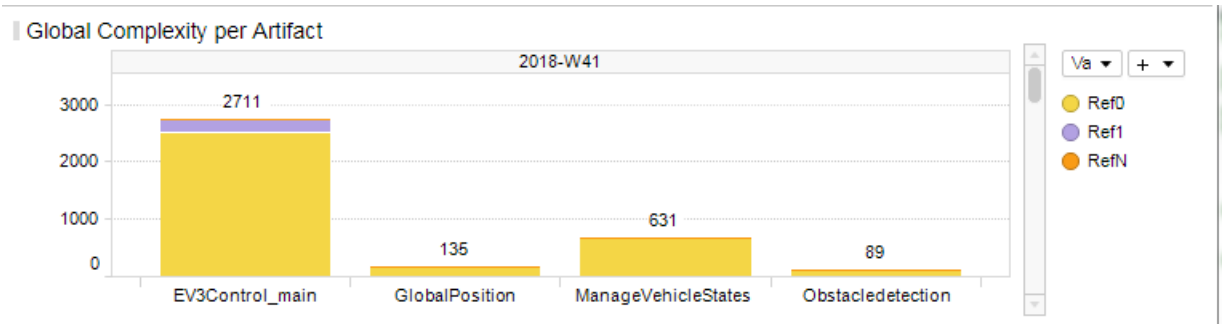


Figure 9.4: MXRAY Global Complexity per Artifact

The Global Complexity visualization (Figure 9.4) shows for each artifact the stacked values of Ref0, Ref1 and RefN, where:

- **Ref0** is the global complexity of the model itself excluding all referenced libraries resp. models
- **Ref1** includes the global complexity of referenced libraries/models exactly one time (each library/model is counted once)
- **RefN** includes the global complexity of referenced libraries/models whereas a complexity value is added for each occurence (each library/model may count multiple times depending on how often a library/model is referenced)

The global complexity visualization shows the values for Ref1 and RefN just in addition to Ref0, which means

- Ref1 (as shown) = Ref1 (as measured) - Ref0
- RefN (as shown) = RefN (as measured) - Ref1

Figure 9.5 shows reported results per base measure per artifact. The user gets an overview on how many subsystems per artifact are stated good, acceptable or bad for all metrics provided by the imported report:

- `Local Complexity`
- `Level`
- `CloneGroups` (`Good` and `Bad` only, see adaptation of clone group values as previously described)

Note: As long as data for multiple revisions is imported and no particular revision is marked respectively, all status visualizations (bar charts) offer the option to scroll between revisions.

The third main visualization showing the trend of subsystem results (see Figure 9.6) shows the same metrics as listed above, but rather its evolution over time than the status for a certain revision.

With that the user is able to see the number of good subsystems that increase during the project runtime while the number of acceptable and bad subsystems decreases. It is also possible to detect an increase of the overall number of subsystems per model at a certain point in time (e.g. if a model was extended during development).

If there are multiple selected artifacts (or no specific one), the user is able to scroll between artifacts to see the trend figures for a particular artifact.

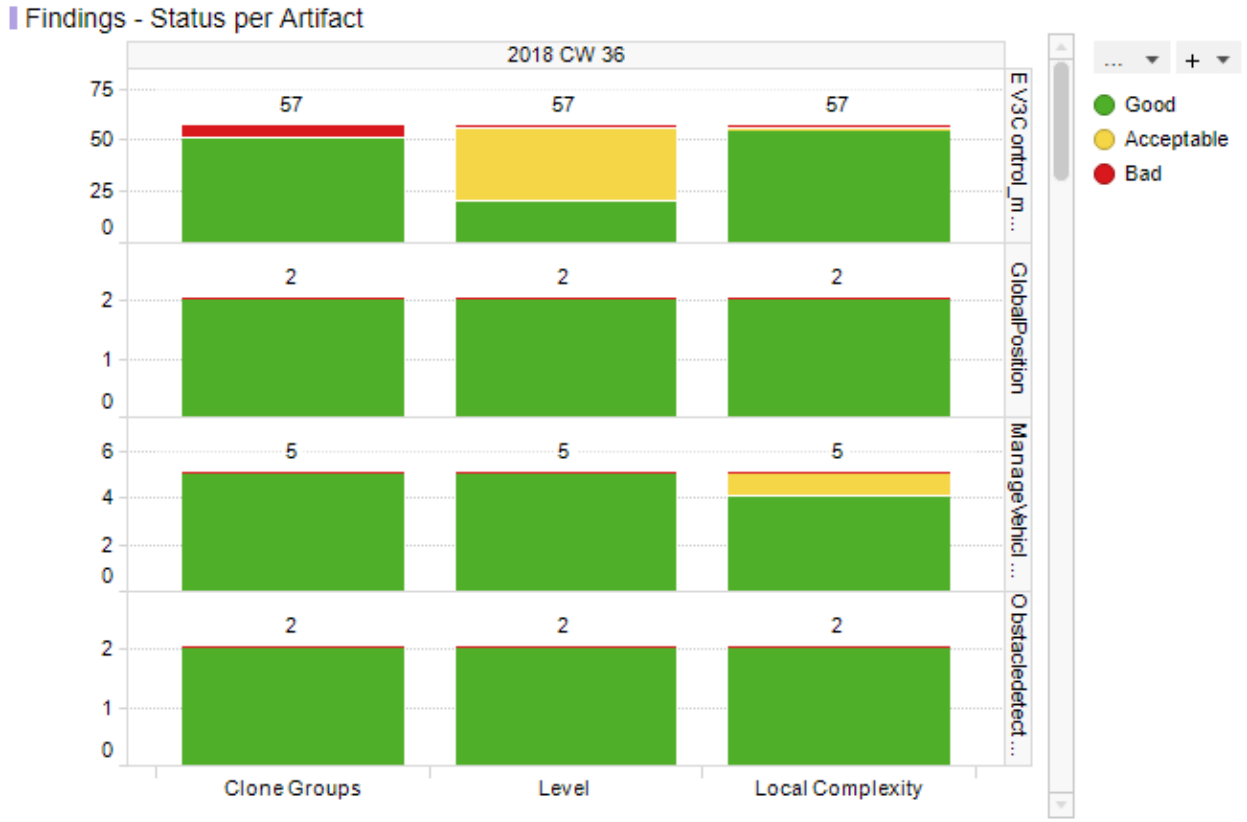


Figure 9.5: MXRAY Subsystems Results Status per Base Measure per Artifact

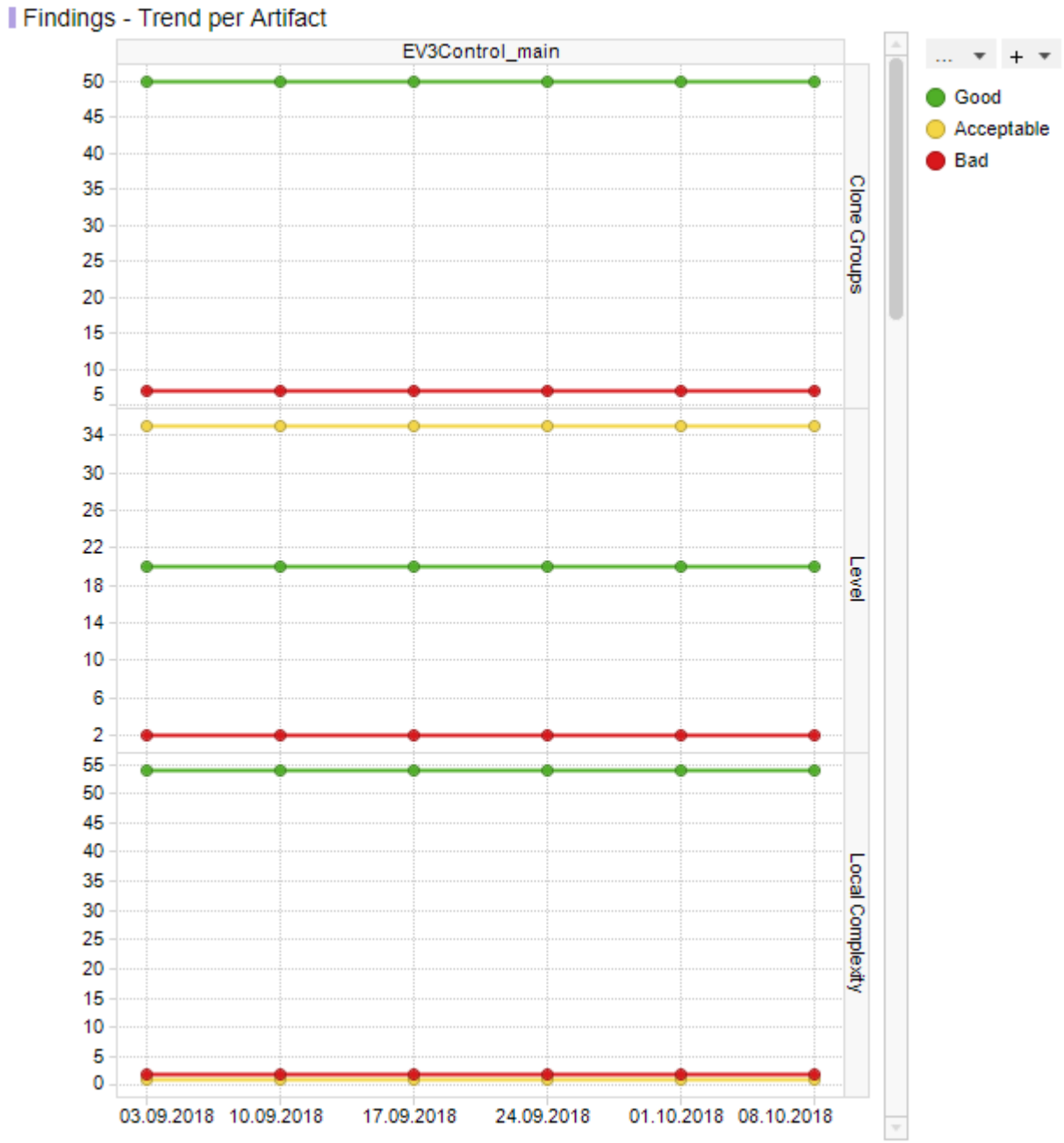


Figure 9.6: MXRAY subsystem results Trend per Base Measure per Artifact

The Artifact KPI on the left-hand side shows a tile for each artifact colored according to the number of found severe issues or issues, respectively.

- **Severe Issues:**

  Sum of all `Bad` measures per artifact.

- **Issues:**

  Sum of all `Bad` AND `Acceptable` measures per artifact.

If there is any Severe Issue, which means a base measure (e.g. `Level` or `Local Complexity`) with a variable value of `Bad` > 0, the artifact tile is colored red.

If there are no Severe Issues, but Issues (which means a base measure with a variable value of `Acceptable` > 0) are found, the artifact tile is colored yellow.

All other artifact KPIs are colored green.

The Project KPI chart on the top-left of the page shows the color of the worst result over all artifacts as well as the overall number of Severe Issues and Issues for the whole project.

The distribution chart (top-right) shows for each revision the number of artifacts binned according to their worst result.

By these means the user gets an overview on

- the global complexity of each artifact compared to all other artifacts
- how many issues (e.g. subsystems with problems) per artifact exist
- which issues per artifact exist (e.g. subsystems with bad local complexity)
- what are the worst issues to concentrate on first (for each artifact and for the whole project)
- and how this has evolved over time (e.g. from project start time until actual date).

9.4.3 MTest Tool Page

To dig more into details from the Base Measures and Quality properties of the MTest tool, MQC provides an additional MTest Details page.

From the menu bar choose **Tools/MQC Tool Pages/Add MTest Details Page** to add the MTest tool page.

The visualized data is structured by

- Requirements
- Test Sequences
- Assessments and
- (Structural) Coverage.

For each Base Measure MTest provides two values, read and visualized by MQC:

- `Total`, e.g. the total number of reviewed Test Sequences, and
- `Reached`, e.g. the actual number of reviewed Test Sequences

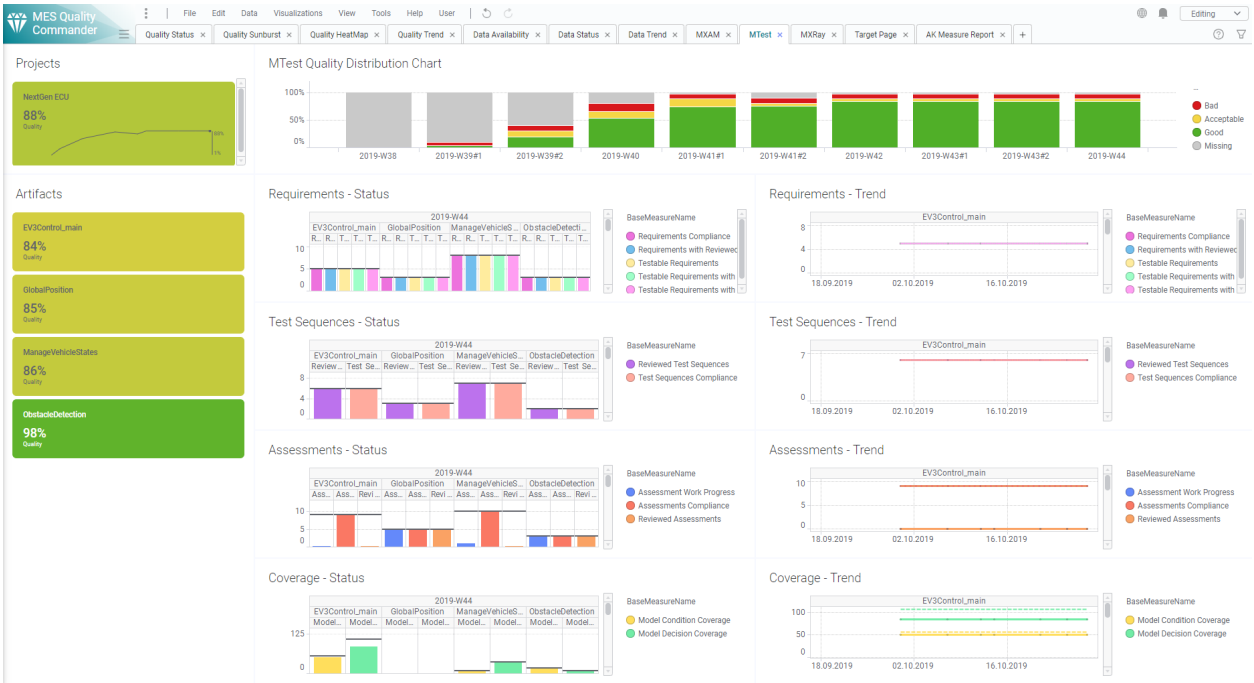


Figure 9.7: MTest Details Page showing Trend and Status

Both values are shown in Status (see Figure 9.8) as well as in Trend (see Figure 9.9).

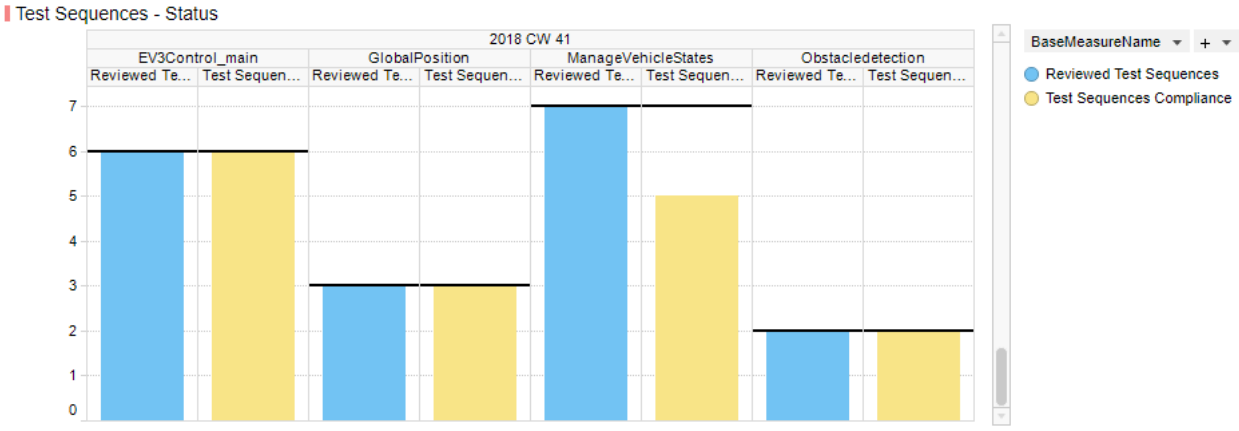


Figure 9.8: Measure Status visualization for Test Sequences

Each status diagram shows the `Reached` values of the imported base measures as bar charts per artifact. The `Total` values are shown as horizontal line (i.e. like a target to be reached) for each measure bar.

As long as data for multiple revisions is imported and no particular revision is selected respectively, MQC offers to scroll between revisions. The *Artifact* KPI selector (on the left-hand side) is used to limit the shown data for particular artifacts.

Each trend diagram shows the `Reached` values of the imported base measures as a trend line over revisions per artifact. The `Total` values are shown as an additional dotted line (i.e. like a target to be reached) for each shown measure trend.

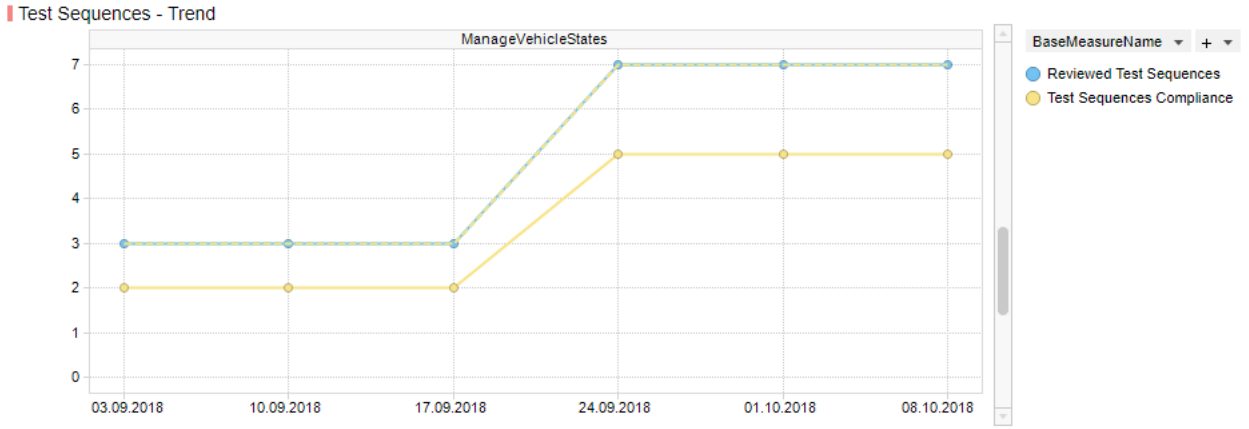


Figure 9.9: Measure Trend visualization for Test Sequences

If multiple artifacts (or no specific artifact) are (is) selected, the user is able to scroll between artifacts to see the trend figures for a particular artifact.

Artifact KPI selector (left-hand side), *Project* KPI selector (top-left) and *Bin Distribution* chart (top-right) are showing quality as on the MQC quality pages.

By these means the user gets an overview on

- the progress of the functional tests
- the current status of the functional tests
- certain Artifact(s)
- a certain test area.

9.4.4 Polyspace Tool Page

MQC provides an additional tool page showing details on data provided by Polyspace.

From the menu bar choose **Tools/MQC Tool Pages/Add Polyspace Details Page** to add the Polyspace tool page.

This page contains four main visualizations:

- Run-Time Checks per Artifact - Absolute
- Run-Time Checks per Artifact - Normalized
- MISRA-C Checks Findings per Artifact
- Global Variables per Artifact

The first Run-Time Checks visualization (Figure 9.11) - top-left of the main visualization area - shows for each artifact the stacked absolute values of all findings excluding the number of green checks:

- `Run-Time Checks.Major` (Number of Red Checks)
- `Run-Time Checks.Moderate` (Number of Orange Checks)

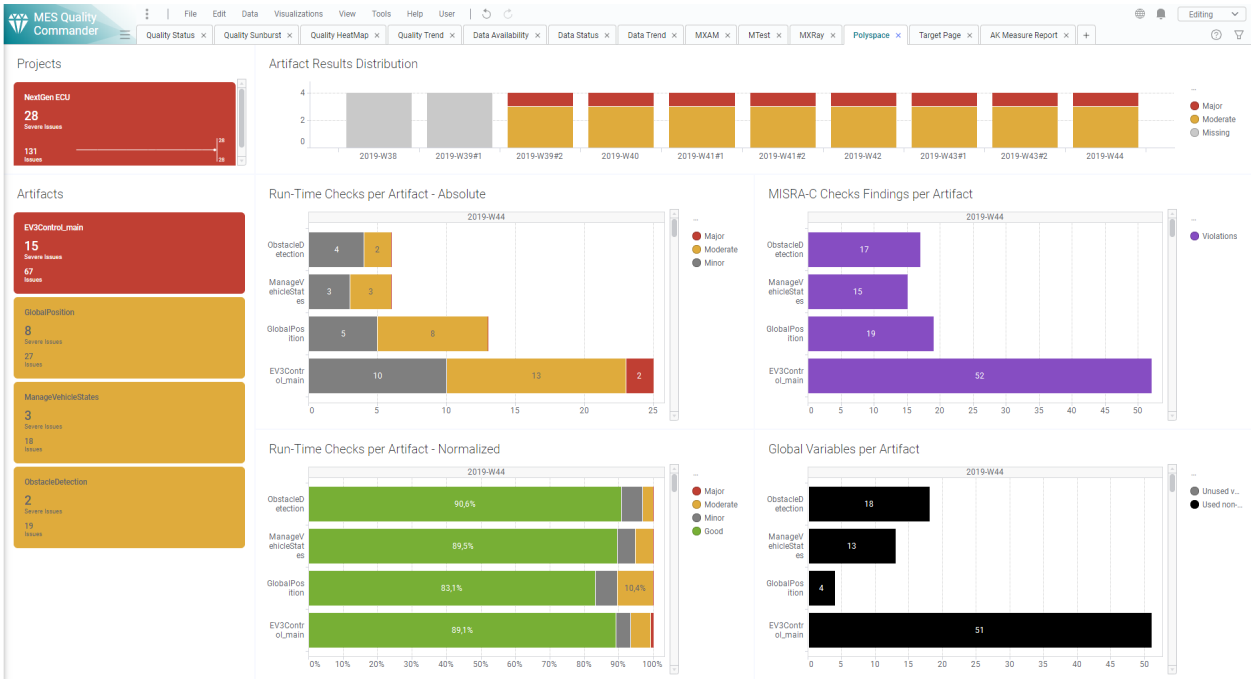


Figure 9.10: Polyspace Details Page

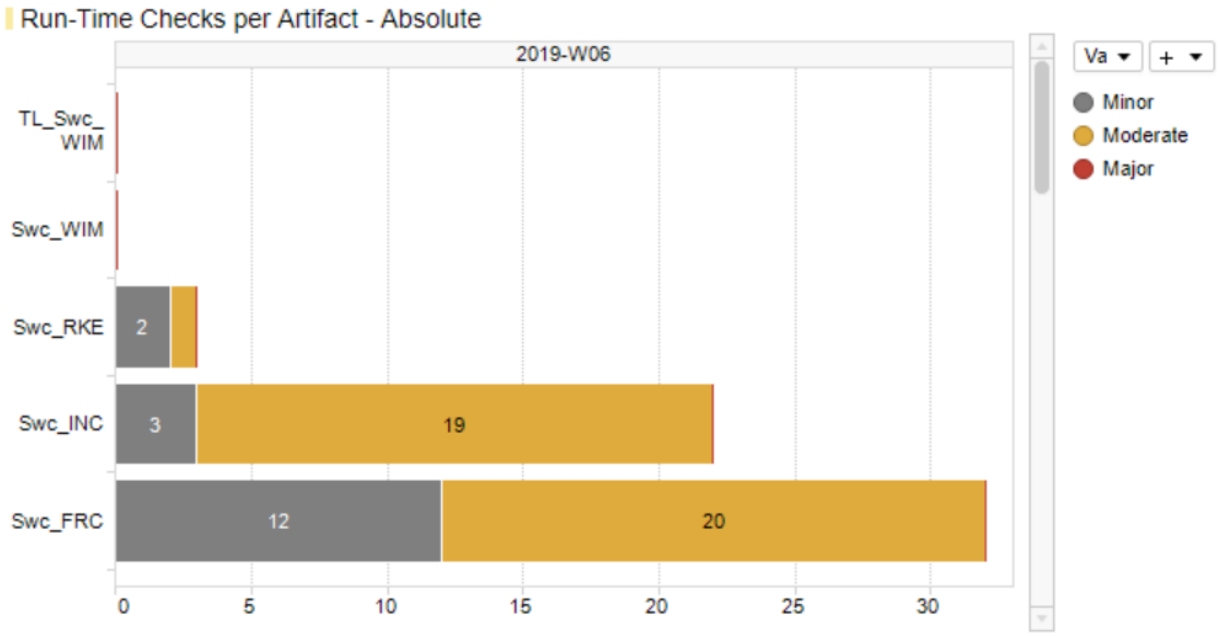


Figure 9.11: Polyspace Run-Time Checks per Artifact - Absolute

- `Run-Time Checks.Minor` (Number of Gray Checks)

This directly indicates the amount of still open and to be solved issues per artifact without being concealed by a huge number of `Run-Time Checks.Good`.

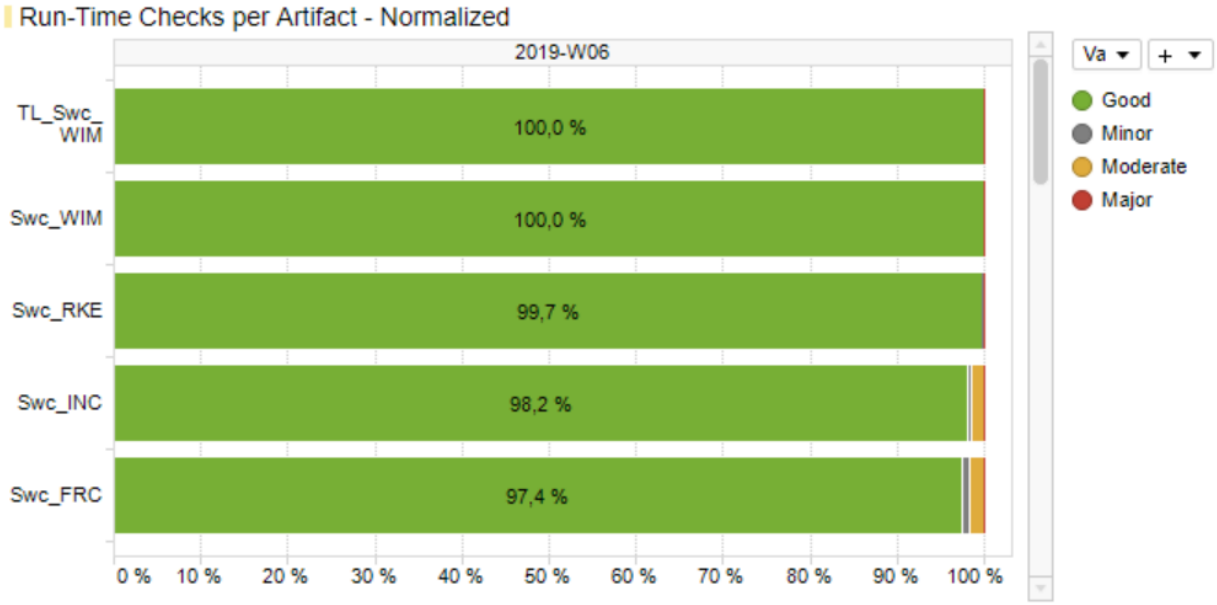


Figure 9.12: Polyspace Run-Time Checks per Artifact - Normalized

Figure 9.12 - at the bottom-left - also shows Run-Time Checks, but this time the relative share (in percent) of each result including the number of green checks compared to the overall number of findings per artifact.

The third main visualization (Figure 9.13) - top-right - shows the absolute number of coding rule violations per artifact.

If a Polyspace report contains different types of coding rule checks, e.g. additionally custom rules, all violations found per artifact will be already summed up by the Polyspace adapter to `MISRA-C Checker.Violations` and will be shown as one value on the MQC data pages as well as on the Polyspace tool page.

The forth main visualization (Figure 9.14) - bottom-right - shows for each artifact the stacked absolute numbers of global variables:

- `Unused variable` (grey)
- `Used non-shared variable` (black)

As long as data for multiple revisions is imported and no particular revision is marked respectively selected, all main visualizations offer to scroll between revisions to get the Polyspace results for a certain point in time.

The Artifact KPI on the left-hand side shows a tile for each artifact colored according to the number of found issues or severe issues, respectively.

- **Severe Issues:**

  Sum of all `Run-Time Checks.Major` and `Run-Time Checks.Moderate` findings per artifact.

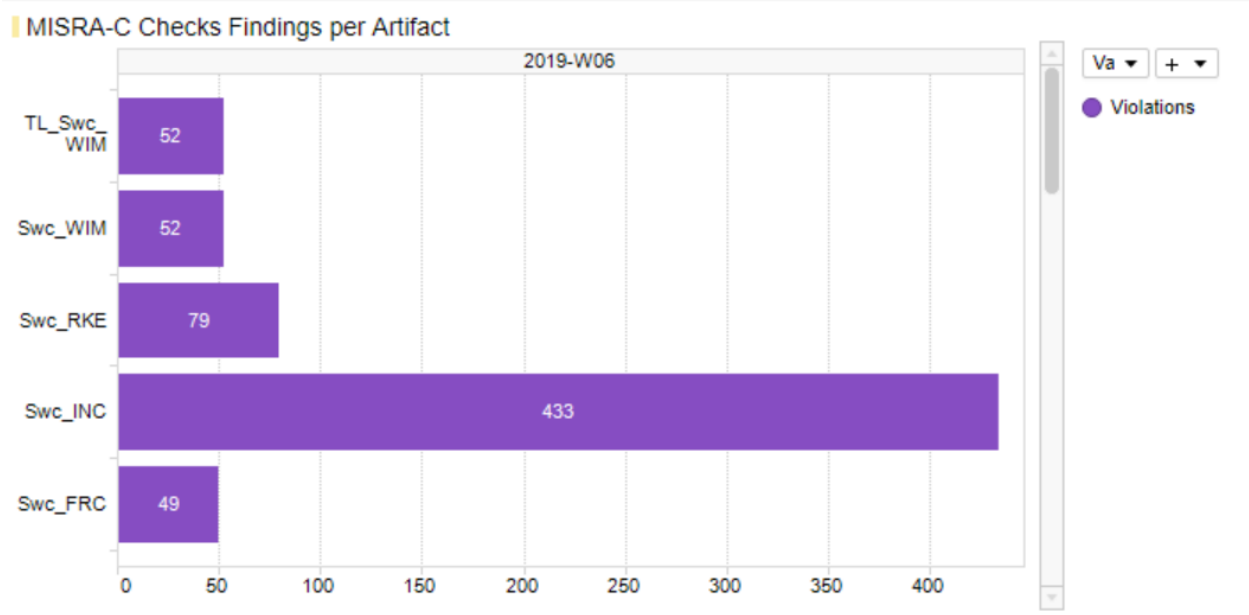


Figure 9.13: Polyspace Coding Rule Violations per Artifact

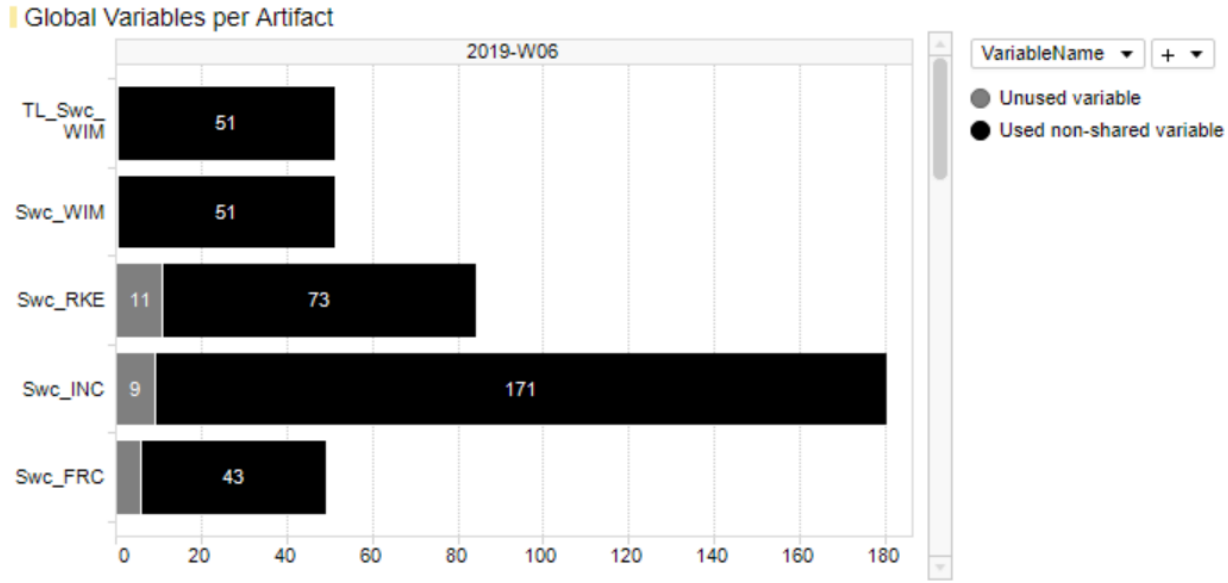


Figure 9.14: Polyspace Global Variables per Artifact

- **Issues:**

  Sum of all `Run-Time Checks.Major`, `Run-Time Checks.Moderate` and `MISRA-C Checker.Violations` findings per artifact.

If there is any `Major` finding, the artifact tile is colored red.

If there is no `Major` but `Moderate` finding, the artifact tile is colored orange.

If there are no Severe Issues at all, but Issues (which means `Violations` > 0), the artifact tile is colored violet.

All other artifact KPIs are colored green.

The Artifact KPI on the left-hand side shows a tile for each artifact colored according to the number of found Severe Issues and Issues respectively. For the example in Figure 9.3 this means that within the current (last) revision there are four artifacts in total, one of them with severe issues (red bin), one with issues only (yellow bin) and two artifacts with no issues at all (green bins).

The Project KPI chart on the top-left of the page shows the color of the worst result over all artifacts as well as the overall number of Severe Issues and Issues for the whole project.

The distribution chart (top-right) shows for each revision the number of artifacts binned according to their worst result.

9.5 COLOR SCHEMES

On Quality Pages (i.e. Quality Status page, Quality Sunburst page and Quality Trend page), MQC uses the traffic light color scheme:

	Good	All quality properties with an evaluated quality between 80% - 100%
	Acceptable	All quality properties with an evaluated quality between 20% - 80%
	Bad	All quality properties with an evaluated quality between 0% - 20%
	Missing	Quality cannot be evaluated because of missing quality measure values

Gradient Coloring	Description	
	100% quality	A gradient coloring is mainly used for certain elements that have a computed or aggregated quality value assigned (e.g. for projects, artifacts and quality properties)
	50% quality	
	0% quality	
	No quality value available	

On Availability Pages (i.e. Data Status page and Data Trend page) the MQC uses a blue coloring scheme. This means you can easily distinguish between available data in blue and missing data in grey.

Categorical Coloring	Description
Available	Measure data is available for a certain revision
Default	Measure value was missing, hence filled up with a previously defined default value
Propagated	Measure was missing, but could be propagated from previous revision
Missing	Measure data is missing for a certain revision

Gradient Coloring	Description	
	100% data available	A gradient coloring is mainly used for certain elements that have an aggregated availability value assigned (e.g. projects and artifacts)
	0% data available	

9.6 MARKING

The purpose of marking is to restrict the shown data to the interesting parts.

The following should be considered while using marking:

- When marking one or multiple elements of a visialization, the data that corresponds to other, not marked, elements are excluded from all visualizations.

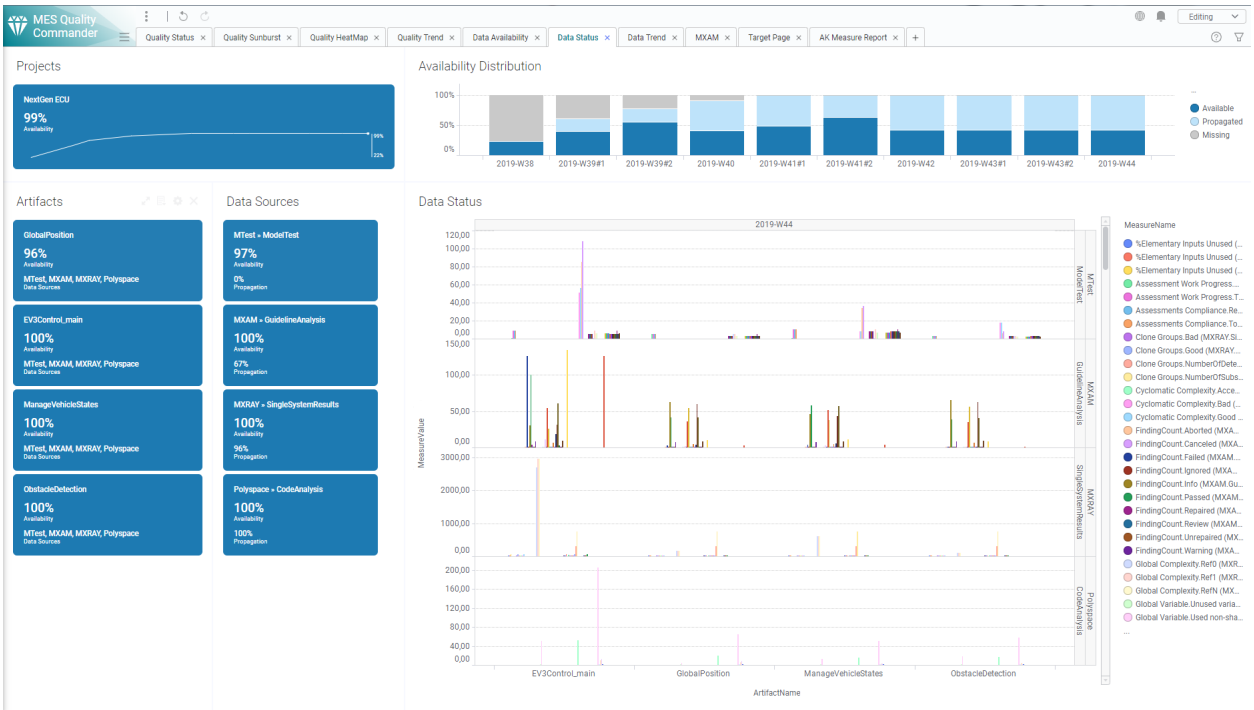


Figure 9.15: Data Status page without any marking

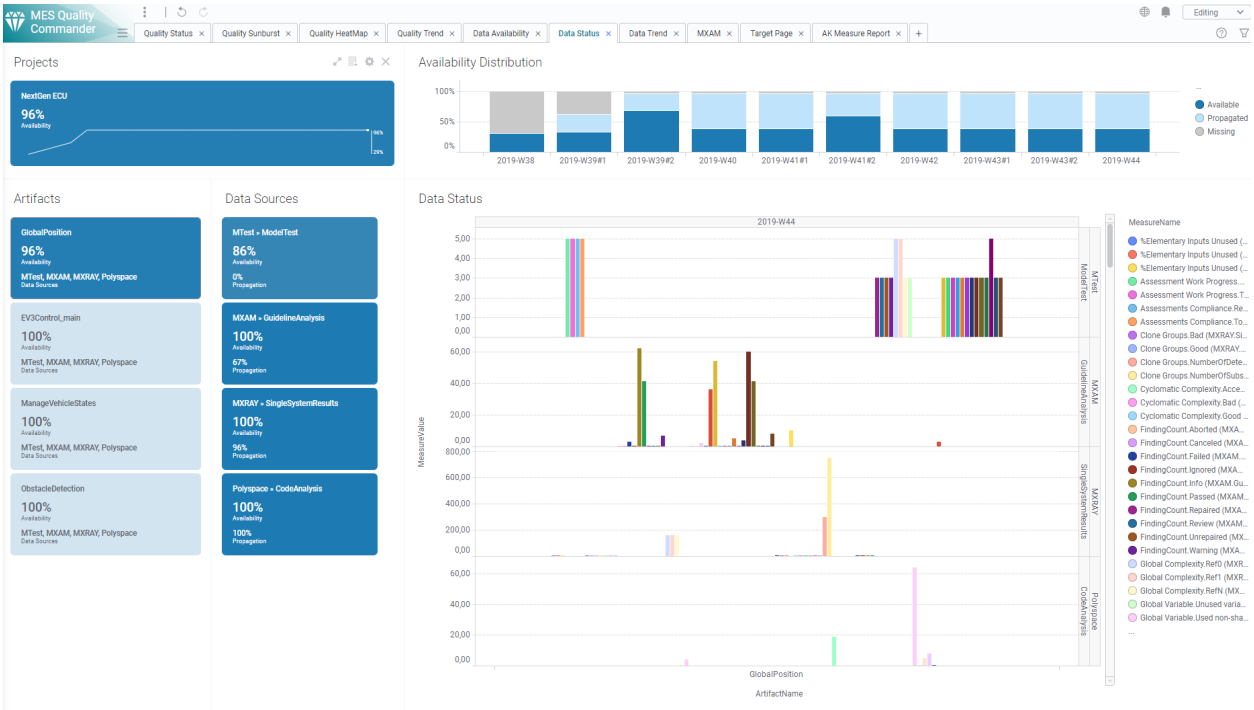


Figure 9.16: Using artifact marking on Data Status: All other visualizations show only data related to the selected artifact (GlobalPosition)

As seen in Figure 9.16, by marking data in one visualization except the main visualization, only data representing this marking selection is shown in each of the other visualizations.

- Please note that marking works cumulative. That means further marking without resetting will reduce the resulting set of data even more. To mark multiple elements in the same visualization, press and hold *Ctrl* before clicking on another element. It is also possible to draw a rectangle enclosing the interesting parts by clicking and holding down the mouse button. (e.g. in Distribution charts).
- Resetting the markings can be achieved by clicking on the Project KPI tile.
- On Status pages (Data Availability, Data Status, Quality Status, Quality Sunburst, Quality Heatmap), a revision is marked with just one click in the distribution visualization. With a second click the quality bin is marked.
- On trend pages (Data Trend, Quality Trend) quality bins can be marked for all revisions with one click.

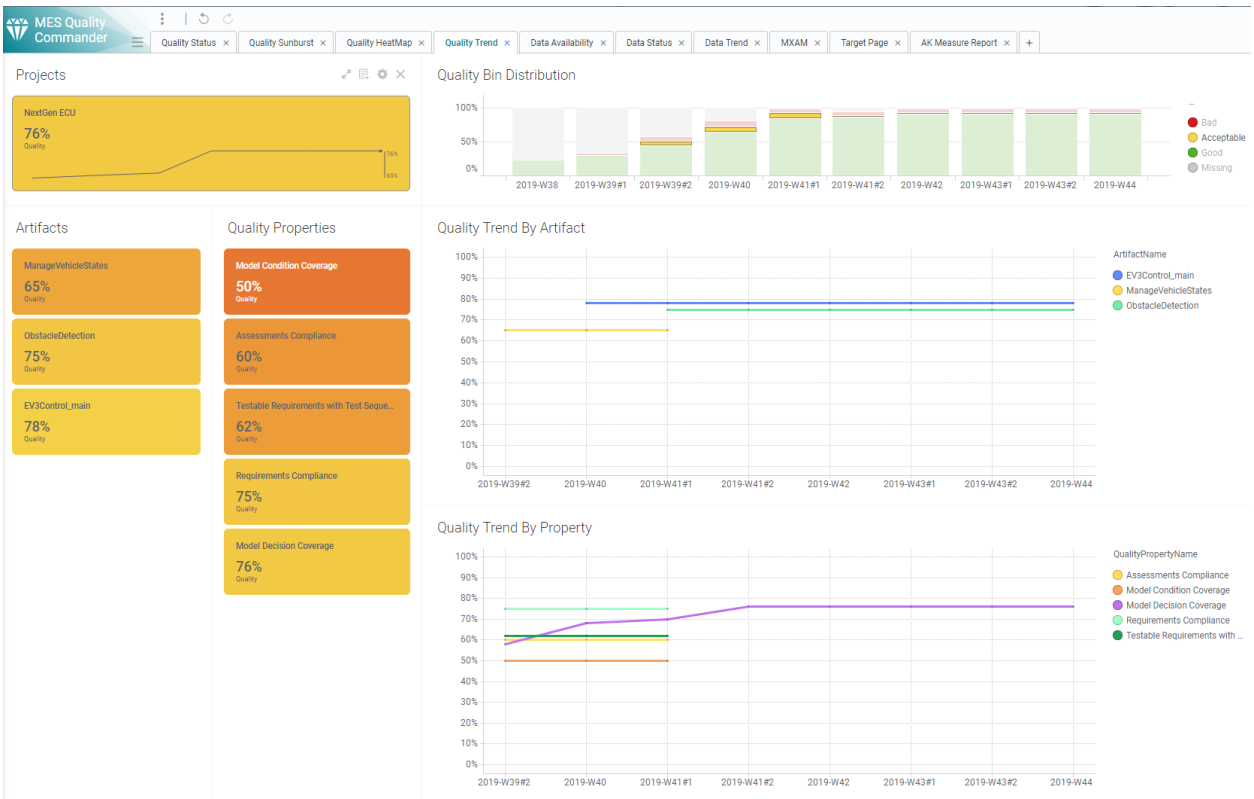


Figure 9.17: On the Quality Trend page the first click on an quality bin will select the quality bins for all revisions

- Artifact and revision marking on quality or data pages will result in the marking on other quality and data pages.

Figure 9.18: Quality Status page after marking *GlobalPosition* on data pages (see Figure 9.16)

10 PROJECT STRUCTURES CONFIGURATION

As explained in *Dimensions and Structures*, the aim of MQC is to visualize project data in a structured way. You have to configure your project in Excel to gain the described possibilities to structure your data.

From the Configuration section of the menu choose **Project Structures/Export**, which provides you with a Project Structures Excel file that can be used as a template and where you can define your own project structures.

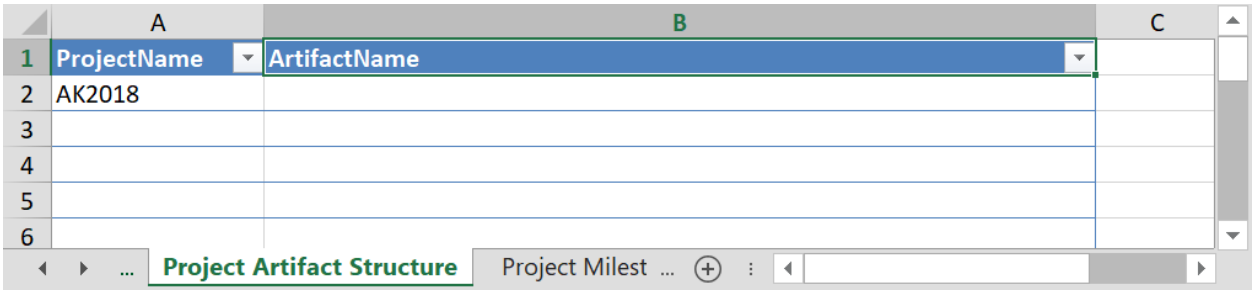
The result file has to be imported again (via **Project Structures/Load**) and the contained configuration will be applied by MQC.

If your imported project structures change, you can easily reload an adapted Project Structures file via **Project Structures/Reload**. Hovering over the `Reload` button provides a tooltip, which helps you detecting name and filepath of the currently imported Project Structures file.

The following sections provide information on how to configure structures in MQC.

10.1 PROJECT NAMING

You can define a dedicated project name and assign artifacts to this project. If the "Project Artifact Structure" does not contain any artifact name, MQC will automatically assign all artifacts for which data is collected by data sources to the given project (Figure 10.1).



	A	B	C
1	ProjectName	ArtifactName	
2	AK2018		
3			
4			
5			
6			

Project Artifact Structure | Project Milest ...

Figure 10.1: Naming of the Project 'AK2018', which contains all imported Artifacts

Please note that if the "Project Artifact Structure" sheet is left empty, i.e. not even a project name is defined there, "Default Project" is used by MQC to visualize your imported data.

10.2 ARTIFACT MAPPING

"Artifact Mapping" allows you to adapt artifact names as they are used by data sources to your needs. Especially if different data sources use different denominations for the same artifact, "Artifact Mapping" may be used to define a common artifact name. All imported measure values collected by the data sources are automatically assigned to the new artifact name.

The artifact name as defined by the "Artifact Mapping" then must be used for all other structuring possibilities, too, i.e. on the "Project Artifact Structure" and the "Artifact Structure" sheet of the Project Structures file.

A	B	C	D	E
ArtifactPath	ArtifactName			
EV3Control_demo_ec	EV3Control_main			
EV3Control_demo_ec/EV3Control	EV3Control_main			
EV3Control_demo_ec/VehicleManager/ObstacleDetection	Obstacledetection			
EV3Control_demo_ec/VehicleManager/VehicleControl/GlobalPosition	GlobalPosition			
GlobalPosition_demo_ec	GlobalPosition			
GlobalPosition_demo_ec/GlobalPosition	GlobalPosition			
ManageVehicleStates_demo_ec	ManageVehicleStates			
ManageVehicleStates_demo_ec/ManageVehicleStates	ManageVehicleStates			
ObstacleDetection_demo_ec	Obstacledetection			
Obstacledetection_demo_ec	Obstacledetection			
Obstacledetection_demo_ec/ObstacleDetection	Obstacledetection			
ObstacleDetection_demo_ec/ObstacleDetection	Obstacledetection			

ContextCategory | ProjectArtifactStructure | ProjectMilestoneStructure | ArtifactStructure | **ArtifactMapping**

Figure 10.2: Map different artifact denominations to common artifact name using "Artifact Mapping" in Project Structures file

Please note that in case the same base measures were provided by different data sources for the same revision and are now mapped to a common artifact name, MQC only uses the most recent of these base measures to calculate quality for the artifact.

10.3 PROJECT MILESTONE STRUCTURE

The "Project Milestone Structure" sheet of the Project Structures file allows you to group your data according to the milestone(s phases) defined for the project. Milestones are assigned to a certain project name and are provided each with a dedicated start date and due date.

In MQC, imported data automatically is assigned to revisions based on the timestamp the data was collected. Multiple revisions are clustered and linked to a particular milestone according to the start date of each revision and the start and due dates defined for the milestones (see Figure 10.3).

The standard way to define your milestones is specifying just a project start date, which is the start date of the first milestone, in combination with a duration for each milestone. Figure 10.4 shows the general way to configure milestones:

Note: In case your project milestone configuration contains a milestone phase that does not fit to your

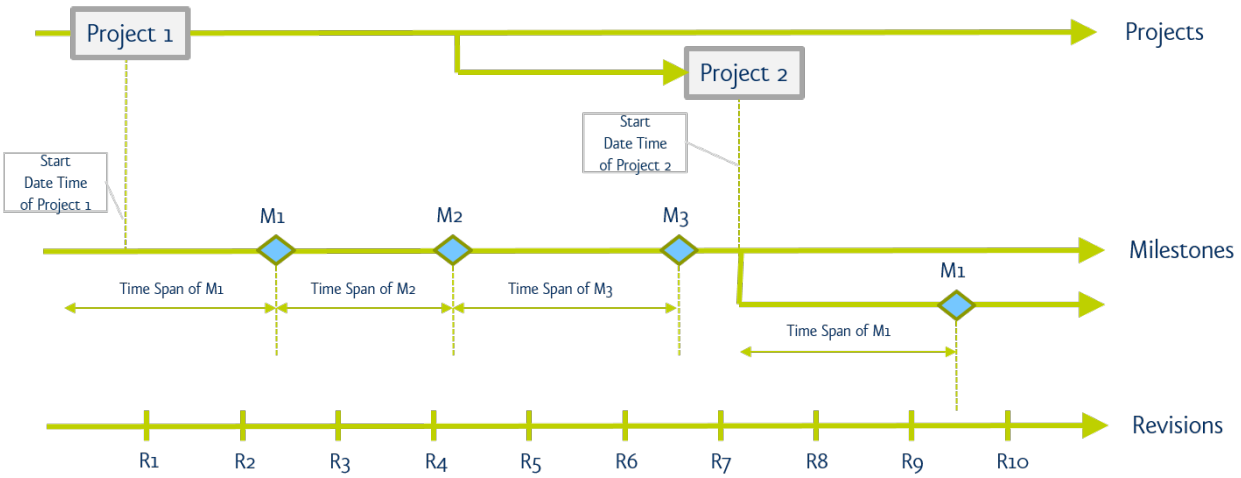


Figure 10.3: Relationship between projects, their milestones and revisions

ProjectName	MilestoneName	MilestoneStartDate	MilestoneDueDate	MilestoneDuration
AK2018	Static Analysis Start	30.08.2018		7
AK2018	Module Test Start			17
AK2018	Static Analysis Full			3
AK2018	Model Test			12
AK2018	This is the Day			7

ContextCategory | ProjectArtifactStructure | **ProjectMilestoneStructure** | ArtifactStructure | ArtifactMapping

Figure 10.4: Definition of milestones for your project using the "Project Milestone Structure" sheet of the Project Structures file with milestone duration specification

imported data, e.g. there are more milestones defined than data available, you can chose if MQC should include these (empty) revisions into your project. By default MQC hides all empty revisions. Nevertheless it offers a functionality to display these configured (but empty) revisions for all artifacts and measures ever found within this project. By clicking on the checkbox `Show Empty Revisions` (to be found within the Revision Granularity section of the Configuration Panel on the left side, see *Revision granularity*) all revisions are filled up with missing data.

Another way of defining your milestones, is by specifying dates into the columns `MilestoneStartDate` and `MilestoneDueDate`. Each milestone needs a start date and a due date, which can be directly set by the user. For example, all revisions with a start date equal to or later than the 6th of September 2018 and before 22nd of September 2018 are linked to the Milestone "Module Test Start" (see Figure 10.5).

	A	B	C	D	E
1	ProjectName	MilestoneName	MilestoneStartDate	MilestoneDueDate	MilestoneDuration
2	AK2018	Static Analysis Start	30.08.2018	05.09.2018	
3	AK2018	Module Test Start	06.09.2018	22.09.2018	
4	AK2018	Static Analysis Full	23.09.2018	25.09.2018	
5	AK2018	Model Test	26.09.2018	07.10.2018	
6	AK2018	This is the Day	08.10.2018	14.10.2018	
7					

... ProjectMilestoneStructure | ArtifactStructure | ...

Figure 10.5: Definition of milestones for your project using the "Project Milestone Structure" sheet of the Project Structures file with date specification

Additionally, MQC is able to calculate missing milestone configuration data based on a given milestone duration, whereas the following order is applied:

- the duration and a given due date is used to calculate a missing start date
- if no due date is given, the duration and a given start date is used to calculate the missing due date
- if neither start date nor due date is given for a certain milestone, the start date is calculated based on the due date of the previous milestone

Missing milestone configuration data is automatically calculated when importing the corresponding structures configuration file.

Note: A proper milestone configuration is essential when working with target values.

10.4 ARTIFACT STRUCTURE

If you would like to group your artifacts hierarchically as described in *Artifacts and Artifact Structure*, this can be done by using the "Artifact Structure" sheet. It allows you to define e.g. groups of different functionalities and to assign your artifacts to these groups. This offers you the possibility to filter the visualized data accordingly, e.g. to just focus on artifacts belonging to a certain group.

	A	B	C	D	E	F
1	ArtifactName	ContextCategories	ArtifactWeight	StructureRoot	StructureGroup	StructureElement
2	EV3Control_main	EV3_CC	1		Model	Main
3	ObstacleDetection		1		Model	Function
4	GlobalPosition	Global_CC	1		Model	Function
5	ManageVehicleStates		1		Model	Function
6						

ProjectArtifactStructure | ProjectMilestoneStructure | **ArtifactStructure** | ArtifactMapping

Figure 10.6: Group artifacts according to real project structures for example according to different functionality

As shown in the previous figure, you can use the columns *StructureRoot*, *StructureGroup* and *StructureElement* of the Excel template to structure your artifacts with respect to e.g. functionality, responsibility, release, internal development process. In theory, you can specify any relationship you want to see for your data. The hierarchy that is spanned in MQC is equal to:

- StructureRoot
  - StructureGroup
    - StructureElement
      - Artifact 1
      - Artifact 2
      - ...

Note: Configuring an artifact structure is optional. Leaving the corresponding columns empty, results in a flat list of artifacts within MQC.

Artifact structuring does not influence quality calculation.

For the explanation of the Context Categories tab please refer to *Define your Context Categories*.

10.5 ARTIFACT WEIGHTS

On the other hand within the "Artifact Structure" sheet the user is able to define artifact weights, which means that some artifacts can have more influence than others within the same project. Hereby, individual artifact parameters can be adapted with regard to the user's interpretation of their importance, reliability, etc. For example, in our example from the *Quick Start Guide* chapter the importance of one artifact has to be considered higher due to the fact of being the main model and therefore should also get a bigger influence within the calculation of quality.

If no artifact weight is defined, MQC will assume equal "size" of all artifacts and assigns a default value of 1 to all listed elements, as shown in the following figure.

ArtifactName	ContextCategories	ArtifactWeight
EV3Control_main	MXAM, MTest	3
Obstacledetection		0.5
GlobalPosition	MXRAY	0.5
ManageVehicleStates	MXAM, MTest, MXRAY	0.5

ProjectArtifactStructure | ProjectMilestoneStructure | **ArtifactStructure** | ArtifactMapping

Figure 10.7: For the main Artifact a weight of 3 (and for others a value of 0.5) is defined and the configuration file loaded as project structure via the Configuration section of the menu

Figure 10.8: *Quality Heatmap* page showing a project with different Artifact "sizes" / weights

Figure 10.9: *Quality Heatmap* page showing a project with equal Artifact "sizes" / weights

11 QUALITY MODEL CONFIGURATION

As described in *Quality Concept*, MQC handles the imported data in such a way that quality can be structured and computed out of it. The configuration of the quality model including the definition of measurement functions to calculate measures for quality properties as well as the definition of base measures and derived measures have to be done in Excel. You can use the Quality Model Excel file provided by MQC as base for your configuration.

The quality model has to be imported (open the **Quality Model** dialog in the Configuration section of the menu and select the `Load` button) and the chosen configuration will be applied by MQC.

If you do not have a proper Quality Model Excel file that can be adapted, you can create one with the help of MQC. From the **Quality Model** dialog choose the `Export` button to export the quality model currently used by MQC (which is the Default Quality Model, if no other quality model was imported before). This provides you an Excel file that can be used as template and where you can define your own project's quality model.

If your imported quality model is changed, you can easily reload an adapted quality model via the `Reload` button. Hovering over the `Reload` button provides a tooltip, which helps you detecting name and filepath of the currently imported Quality Model file.

The following sections provide information on how to configure the quality model and the intended quality structures in MQC.

11.1 BASE MEASURES AND DEFAULT VALUES

The "Base Measure" sheet of the quality model is used to define all base measures that you expect to be imported from a certain data source.

As described later in *Derived Measures* and *Quality Properties*, these base measures are used to calculate other measures, so-called derived measures, and – most important – quality measures, which are the base to compute quality.

For configuring base measures, MQC expects the following structure:

- data source name
- (default) measurement name
- base measure name
- variable name

- default value

The data source is the tool, which is used to provide data to be imported into MQC. Examples are MXAM and MTest.

Typically, a measurement means the set of operations done to determine the value for a measure. For example this can be the name of a specific guideline document, which is used for the static analysis of a model, or the kind of test environment like MiL, SiL or PiL.

The measurement name configured in the "Base Measure" sheet is used as default, if a data source does not provide a measurement name. If a data file contains a measurement name, the default is ignored and MQC uses the name provided by the data source.

Additionally, the "Base Measure" sheet provides you the possibility to define a default value per base measure variable. Please remember that base measures of the same type may be grouped (refer to *Measures and Measurements*), which results in a base measure containing multiple variable-value pairs.

Without default values, in case there are values missing for variables inside such a base measure group for a certain revision, a measurement function using these missing base measure variables could not be calculated. Therefore, MQC uses a configured default in case the data source has provided values for some base measure variables of a base measure group but not for all.

As an example, please imagine the base measure group `FindingCount`, which contains the variables `Canceled`, `Aborted`, `Review`, `Failed`, `Unrepaired`, `Warning`, `Info`, `Repaired`, `Passed` and `Ignored`. If for a certain revision the data source does not provide values for e.g. `Repaired` and `Passed`, MQC uses the default values for the related base measure variables as defined in the "Base Measure" sheet (see Figure 11.1). Hence, a corresponding measurement function can be calculated.

	A	B	C	D	E
1	DataSourceName	MeasurementName	BaseMeasureName	VariableName	DefaultValue
2	MXAM	GuidelineAnalysis	FindingCount	Aborted	0
3	MXAM	GuidelineAnalysis	FindingCount	Canceled	0
4	MXAM	GuidelineAnalysis	FindingCount	Failed	0
5	MXAM	GuidelineAnalysis	FindingCount	Info	0
6	MXAM	GuidelineAnalysis	FindingCount	Passed	0
7	MXAM	GuidelineAnalysis	FindingCount	Repaired	0
8	MXAM	GuidelineAnalysis	FindingCount	Review	0
9	MXAM	GuidelineAnalysis	FindingCount	Unrepaired	0
10	MXAM	GuidelineAnalysis	FindingCount	Warning	0
11	MXAM	GuidelineAnalysis	FindingCount	Ignored	0

... DerivedMeasure BaseMeasure Cont ...

Figure 11.1: Defining default values for base measures not provided by a data source

Note: Please note, that in the "Base Measure" sheet of the quality model all expected base measure variables have to be defined. This is to assure that missing values can be substituted with the default values.

In addition, MQC uses default values only if a base measure variable inside a base measure group is missing! If the complete base measure is missing for a revision, no default values will be used!

In that case, a base measure is completely missing in the current revision but values exist for any of the previous revisions, data propagation may be used to gain a full set of available information.

11.2 DERIVED MEASURES

Derived measures are computed either from base measures or from other derived measures as explained in *Derived Measures*.

This may be used to shorten the definition of measurement functions, which compute quality measures for quality properties (see *Quality Properties*).

If you would like to configure derived measures, this can be done using the "Derived Measure" sheet of the Quality Model Excel file.

	A	B	C	D	E
1	DataSourceN	MeasurementName	DerivedMeasureName	VariableName	MeasureFunction
2			GuidelinesCalc	Passed	[GuidelineCount.Passed] + log(1 + [FindingCount.Passed], 2)
3			GuidelinesCalc	Repaired	[GuidelineCount.Repaired] + log(1 + [FindingCount.Repaired], 2)
4			GuidelinesCalc	Unrepaired	[GuidelineCount.Unrepaired] + log(1 + [FindingCount.Unrepaired], 2)
5			GuidelinesCalc	Failed	[GuidelineCount.Failed] + log(1 + [FindingCount.Failed], 2)
6			GuidelinesCalc	Info	[GuidelineCount.Passed with Infos] + log(1 + [FindingCount.Info], 2)
7			GuidelinesCalc	Review	[GuidelineCount.Review] + log(1 + [FindingCount.Review], 2)
8			GuidelinesCalc	Warning	[GuidelineCount.Warning] + log(1 + [FindingCount.Warning], 2)

QualityModel | DerivedMeasure | BaseMeasure | ...

Figure 11.2: Defining derived measures using base measures provided by data sources

A fullqualified derived measure name consists of four parts (same structure as for base measures):

- data source name
- measurement name
- derived measure name
- variable name

With that you are able to define groups of derived measures, each group containing multiple variables. The configuration of data source name and measurement name is optional. If nothing is set here, MQC takes data source and measurement from the base and/or derived measures used to calculate the derived measure.

Figure 11.2 shows the definition of several derived measures that are computed from various variables of the base measures `FindingCount` and `GuidelineCount` provided by the data source MXAM. For example, the derived measure `GuidelinesCalc.Passed` is calculated by the following measure function:

```
[GuidelineCount.Passed] + log(1 + [FindingCount.Passed], 2)
```

Note: As shown in the above example, base and derived measures used in a measure function must be set in square brackets `[` and `]`!

You can define measure functions using a short notation for measure names only consisting of base / derived measure name and variable name (see Figure 11.2). MQC then checks for matching measures that were imported and replaces the used measure names by the fullqualified names.

If the quality model is exported (choose the `Export` button from the **Quality Model** dialog), the "Derived Measure" sheet of the resulting Excle file contains the fullqualified names as shown in Figure 11.3).

	A	B	C	D	E
1	DataSource	MeasurementNam	DerivedMeasureN	VariableName	MeasureFunction
2	MXAM	GuidelineAnalysis	GuidelinesCalc	Passed	[MXAM.GuidelineAnalysis.GuidelineCount.Passed] + log(1 + [MXAM.GuidelineAnalysis.FindingCount.Passed], 2)
3	MXAM	GuidelineAnalysis	GuidelinesCalc	Repaired	[MXAM.GuidelineAnalysis.GuidelineCount.Repaired] + log(1 + [MXAM.GuidelineAnalysis.FindingCount.Repaired], 2)
4	MXAM	GuidelineAnalysis	GuidelinesCalc	Unrepaired	[MXAM.GuidelineAnalysis.GuidelineCount.Unrepaired] + log(1 + [MXAM.GuidelineAnalysis.FindingCount.Unrepaired], 2)
5	MXAM	GuidelineAnalysis	GuidelinesCalc	Failed	[MXAM.GuidelineAnalysis.GuidelineCount.Failed] + log(1 + [MXAM.GuidelineAnalysis.FindingCount.Failed], 2)
6	MXAM	GuidelineAnalysis	GuidelinesCalc	Info	[MXAM.GuidelineAnalysis.GuidelineCount.Passed with Infos] + log(1 + [MXAM.GuidelineAnalysis.FindingCount.Info], 2)
7	MXAM	GuidelineAnalysis	GuidelinesCalc	Review	[MXAM.GuidelineAnalysis.GuidelineCount.Review] + log(1 + [MXAM.GuidelineAnalysis.FindingCount.Review], 2)
8	MXAM	GuidelineAnalysis	GuidelinesCalc	Warning	[MXAM.GuidelineAnalysis.GuidelineCount.Warning] + log(1 + [MXAM.GuidelineAnalysis.FindingCount.Warning], 2)

QualityModel | DerivedMeasure | BaseMeasu ...

Figure 11.3: Derived measures with data source and measurement name set by MQC as well as measure function containing fullqualified measure names

If a measure function contains a base measure that exists for different measurements, for example if `FindingCount.Passed` was measured multiple times by checking compliance according to different guideline documents, MQC calculates the corresponding derived measure for each of those measurements. The same applies to measure functions containing derived measures calculated for multiple measurements.

Note: If a derived measure is calculated for multiple measurements, the configured data source and measurement names will be replaced by data source and measurement from the base and/or derived measures used to calculate the derived measure.

To calculate a derived measure for a selected measurement only, please, use fullqualified measure names when defining the measure function (as shown in Figure 11.3).

11.3 QUALITY PROPERTIES

The "Quality Model" sheet is used to define the lowest level of computable quality namely quality properties. For each quality property a measurement function has to be defined that is used to calculate a quality measure between 0 and 1 by using base and derived measures.

Derived measures (see *Derived Measures*) can be used instead of base measures to simplify quality computation, hence, to shorten the formula of a particular measurement function.

Figure 11.4 shows an example for a measurement function, where all base measure variables for the base measures `FindingCount` and `GuidelineCount` were replaced by the corresponding derived measures.

	D	E	F
1	QualityPropertyName	QualityPropertyWeight	QualityMeasurementFunction
2	Guideline Compliance	1	(1.0 * ([GuidelinesCalc.Passed] + [GuidelinesCalc.Info] + [GuidelinesCalc.Repaired]) + 0.8 * [GuidelinesCalc.Warning] + 0.5 * [GuidelinesCalc.Unrepaired] + 0.2 * [GuidelinesCalc.Failed] + 0.1 * [GuidelinesCalc.Review] + 0.0 * [GuidelinesCalc.AbortedCanceled]) / ([GuidelinesCalc.Passed] + [GuidelinesCalc.Info] + [GuidelinesCalc.Repaired] + [GuidelinesCalc.Warning] + [GuidelinesCalc.Unrepaired] + [GuidelinesCalc.Failed] + [GuidelinesCalc.Review] + [GuidelinesCalc.AbortedCanceled])

QualityModel | DerivedMeasure | BaseMeası ...

Figure 11.4: Reducing measurement functions by using derived measures defined on the "Derived-Measure" sheet

---

Note: As shown in the above example, base and derived measures used in a measurement function must be set in square brackets `[` and `]`!

As for measure functions to calculate derived measures, you can define quality property measurement functions using a short notation for measure names consisting of base / derived measure name and variable name. MQC then checks for matching measures that were imported respectively calculated and replaces the used measure names by the fullqualified names.

If a measurement function uses a base or derived measure that exists for different measurements, for example if `GuidelinesCalc.Passed` was calculated multiple times, MQC calculates the corresponding quality property for each of the measurements.

Note: If a quality property is calculated for multiple measurements, the configured quality property name will be extended by the name of the measurement, e.g. `Guideline Compliance_GuidelineAnalysis`, whereas `Guideline Compliance` is the defined quality property name and `GuidelineAnalysis` is the name of the measurement.

To calculate a quality property for a selected measurement only, please, use fullqualified measure names when defining the measurement function.

Additionally as explained in *Quality Model Structure*, quality in MQC is aggregated on different levels (as shown in Figure 6.4). MQC allows to assign quality properties with their measurement functions to sub-characteristics. Those sub-chracteristics again can be assigned to characteristics (see Figure 11.5). This enables the user to group quality properties in order to define different quality aspects, which may be analysed separately.

The following sunburst visualization (Figure 11.6, lower right) illustrates the computation of the configured quality model.

The outer ring of the sunburst consists of all quality properties defined by the quality model. Please note, each quality property is already aggregated over the artifacts (see Figure 6.6 in section *Quality Aggregation*).

The next ring shows the sub-characteristics defined by the quality model. A sub-characteristic is calculated using the average of all quality properties assigned to that sub-characteristic.

	A	B	C	D
1	QualityModel	QualityCharacteristic	QualitySubCharacteristic	QualityPropertyName
2	Product Quality Model	Correctness	Model Design	Guideline Compliance
3	Product Quality Model	Correctness	Architectural Design	Local Complexity
4	Product Quality Model	Correctness	Architectural Design	Modelling Depth
5	Product Quality Model	Correctness	Architectural Design	ModelClones
6	Product Quality Model	Correctness	Architectural Design	Inports
7	Product Quality Model	Correctness	Architectural Design	Outports
8	Product Quality Model	Correctness	Architectural Design	%Elementary Inputs Unused (globally)
9	Product Quality Model	Correctness	Functional Requirements	Requirements Compliance
10	Product Quality Model	Completeness	Functional Requirements	Requirements with Reviewed Testability
11	Product Quality Model	Completeness	Functional Requirements	Testable Requirements with Assessments
12	Product Quality Model	Completeness	Functional Requirements	Testable Requirements with Test Sequences
13	Product Quality Model	Completeness	Test Sequences	Test Sequence Work Progress
14	Product Quality Model	Completeness	Test Sequences	Reviewed Test Sequences
15	Product Quality Model	Correctness	Test Sequences	Test Sequence Compliance
16	Product Quality Model	Completeness	Assessments	Assessment Work Progress
17	Product Quality Model	Correctness	Assessments	Assessments Compliance

QualityModel | DerivedMeasure | BaseMeasure | ...

Figure 11.5: Define quality properties, measurement functions and quality model structures

Figure 11.6: Computation of the overall quality illustrated using the sunburst visualization.

The third ring of the sunburst chart shows the configured characteristics. Each characteristic is calculated using the average of all quality properties assigned to all sub-characteristics belonging to that particular characteristic.

Finally, the inner circle shows the overall quality.

Note: Per default, MQC uses all quality properties to compute the separate levels of the quality model, even if the measure value for a certain quality property cannot be calculated. This may be the case if all or a few measures, which are used to compute a quality property measurement function, are not available within a certain revision, i.e. not yet measured.

Missing quality properties always are treated as 0 when calculating the overall quality of a project.

MQC allows to exclude missing quality properties for the calculation. To achive this, open the **Settings** dialog in the Configuration panel and enable the `Exclude Empty Values` check box. This may be used to compute quality only for those parts of a project, which are already available, respectively to compute quality only based on those measures already provided.

In the example shown in Figure 11.6 all quality properties defined by the quality model are treated equally, means they have the same weight when computing the overall quality of a project. Nevertheless, MQC allows to give some quality properties a higher importance than others, which is explained in the following section (*Quality Property Weights*).

11.4 QUALITY PROPERTY WEIGHTS

MQC supports to define the relative importance of quality properties. In this way the weight of the process steps is mapped into the quality model. Herewith the result of the quality calculation is representing the process more realistically, similar to the definition of *Artifact Weights*.

In our last example (see Figure 11.6) all defined quality properties had the same weight of 1. Now we use different weights per quality property (Figure 11.7):

	A	B	C	D	E
1	QualityModel	QualityCharacteristic	QualitySubCharacteristic	QualityPropertyName	QualityPropertyWeight
2	Product Quality Model	Correctness	Model Design	Guideline Compliance	2,5
3	Product Quality Model	Correctness	Architectural Design	Local Complexity	2
4	Product Quality Model	Correctness	Architectural Design	Modelling Depth	1
5	Product Quality Model	Correctness	Architectural Design	ModelClones	1
6	Product Quality Model	Correctness	Architectural Design	Inports	0,5
7	Product Quality Model	Correctness	Architectural Design	Outports	0,5
8	Product Quality Model	Correctness	Architectural Design	%Elementary Inputs Unused (globally)	0,5
9	Product Quality Model	Correctness	Functional Requirements	Requirements Compliance	3
10	Product Quality Model	Completeness	Functional Requirements	Requirements with Reviewed Testability	0,25
11	Product Quality Model	Completeness	Functional Requirements	Testable Requirements with Assessments	1,5
12	Product Quality Model	Completeness	Functional Requirements	Testable Requirements with Test Sequences	2
13	Product Quality Model	Completeness	Test Sequences	Test Sequence Work Progress	1,5
14	Product Quality Model	Completeness	Test Sequences	Reviewed Test Sequences	0,5

QualityModel | DerivedMeasure | BaseMeasure | ContextCate ...

Figure 11.7: Quality property weight definition with unequal weights

After replacing the quality model (via the **Quality Model** dialog `Load` button in the Configuration section of the menu), the sunburst visualization adapts as well as the quality measures calculated for sub-characteristics, characteristics and the overall quality of the project.

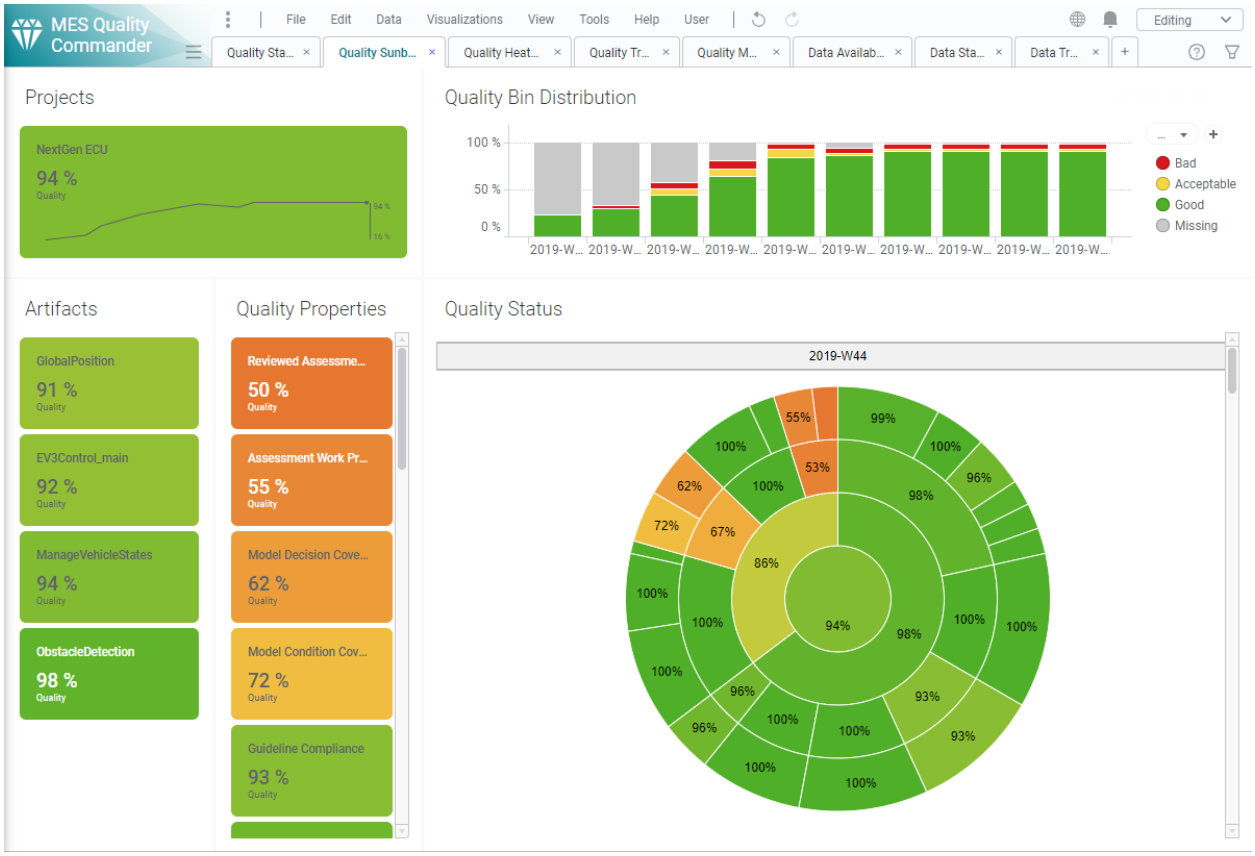


Figure 11.8: All quality properties now have a size according to their configured weights and are included with unequal proportions in the computation of the overall quality.

The size of each quality property tile (outer ring of the sunburst visualization) adjusts according to the weight of the quality property configured in the quality model.

Each level of the quality model now uses the weighted average for quality computation.

Finally, Figure 11.8 shows a different overall quality (inner circle of the sunburst chart) compared to the example with equal weights as shown in Figure 11.6.

11.5 CONTEXT CATEGORIES

By using context categories you can define, which data (on the level of base measures) is expected for a certain artifact. In that way, data never provided and therefore normally stated as “missing” - and probably resulting in bad quality - is treated as “excluded” data. This means it is ignored in availability and quality calculation.

Configuration of context categories is done in two steps:

- assigning data to (or explicitly removing from) a context category is done via the "Context Category" sheet within the quality model configuration file, because this is directly related to the base measures defined here
- assigning context categories to artifacts is done in the project structure configuration file, where the expected artifacts are defined

For a detailed explanation how to configure context categories please refer to chapter *Context Categories*.

11.6 QUALITY MODEL KIT

To help you defining a proper quality model, MQC provides predefined tool-specific quality property measurement functions as part of the installation under the following path: `C:\Users\...\AppData\Local\TIBCO\Spotfire\10.3.3\Modules\MES.MQC.CoreExtension_X.X.X.X\Resources\QualityModel\`. This folder contains Quality Model suggestions for all Data Sources supported by MQC. You will find examples, which allow you to easily assemble your own project-specific quality model (including expected base and derived measures, as well as proposed measurement functions) depending on the data sources you are using within your project.

The following sections will give you examples based on MES data sources, which may help you to define your own measurement functions.

11.7 QUALITY MODEL FOR MES DATA SOURCES

11.7.1 Quality Model for MXAM

In section *Quality Properties* there were already examples shown, how to define derived measures and quality properties, as well as a measurement function using measures provided by MXAM.

The defined derived measures are calculated using both, guideline results (`GuidelineCount`) as well as check findings (`FindingCount`). Because of the fact that the amount of check results typically is much higher than the number of the guideline results, we use a `log` function to reduce the impact of the check findings.

```
GuidelinesCalc.Passed = [GuidelineCount.Passed] + log(1 + [FindingCount.Passed], 2)
```

The MXAM quality model contains derived measures for all possible results, i.e. `Failed`, `Info`, `Warning` and so forth.

Those derived measures are used in the measurement function for the quality property `Guideline Compliance`, which is defined as follows:

```
(1.0 * ([GuidelinesCalc.Passed] + [GuidelinesCalc.Info] + [GuidelinesCalc.Repaired]) +
0.8 * [GuidelinesCalc.Warning] +
0.5 * [GuidelinesCalc.Unrepaired] +
0.2 * [GuidelinesCalc.Failed] +
0.1 * [GuidelinesCalc.Review] +
```

(continues on next page)

(continued from previous page)

```
0.0 * [GuidelinesCalc.AbortedCanceled]) /
([GuidelinesCalc.Passed] + [GuidelinesCalc.Info] + [GuidelinesCalc.Repaired] +
[GuidelinesCalc.Warning] + [GuidelinesCalc.Unrepaired] + [GuidelinesCalc.Failed] +
[GuidelinesCalc.Review] + [GuidelinesCalc.AbortedCanceled])
```

As you can see, the measurement function calculates the average of all findings, whereas each finding gets an individual weight.

There are two additonal MXAM derived measures defined.

`IssueCount.Value` which is the sum of all "bad" findings, i.e.

- `FindingCount.Canceled`
- `FindingCount.Aborted`
- `FindingCount.Review`
- `FindingCount.Failed`
- `FindingCount.Unrepaired`
- `FindingCount.Warning`

`SevereIssueCount.Value` which is the sum of all "bad" findings not considering `FindingCount.Warning`.

Note: Do not remove `IssueCount.Value` and `SevereIssueCount.Value`! The calculation of those values is needed for correct displaying the MXAM tool page (see *MXAM Tool Page*).

11.7.2 Quality Model for M-XRAY

The MXRAY quality model contains quality properties based on base measures provided by the data source M-XRAY.

As an example, the measurement function for `Local Complexity` is defined as follows:

```
(1.0 * [Local Complexity.Good] +
0.8 * [Local Complexity.Acceptable] +
0.2 * [Local Complexity.Bad]) /
([Local Complexity.Good] + [Local Complexity.Acceptable] + [Local Complexity.Bad])
```

The same pattern may be used for `Modelling Depth`, `Level`, `Inports`, `Outports` as well as for `%Elementary Inputs Unused (globally)`.

11.7.3 Quality Model for MTest

The MTest quality properties are defined by comparing current and expected results, which leads to quite simple measurement functions, e.g. the quality property `Requirements with Reviewed`

`Testability` is calculated as follows:

```
[Requirements with Reviewed Testability.Reached] / [Requirements with Reviewed␣
↪Testability.Total]
```

Following this pattern, the MTest quality model additionally defines quality properties for:

- `Requirements Compliance`
- `Testable Requirements with Assessments`
- `Testable Requirements with Test Sequences`
- `Test Sequence Work Progress`
- `Reviewed Test Sequences`
- `Test Sequence Compliance`
- `Assessment Work Progress`
- `Assessments Compliance` and
- `Reviewed Assessments`.

11.8 MEASUREMENT FUNCTION DEFINITION

This section aims to give you an overview on how to define your own quality properties or adapt your measurement functions.

Therefore you can use a bunch of mathematical functions provided within MQC. Apart from the four basic arithmetic operations (`+`, `-`, `*` and `/`), MQC allows to use: `Abs`, `Exp`, `Ceiling`, `Floor`, `Round`, `Ln`, `Log`, `Mod`, `Power`, `Product`, `Sqrt`, `Min`, `Max`, `Avg`, `WeightedAverage`, `Case... when` and `IF... else`.

In the following some of these functions are explained with their possible usage within a measurement function.

We already have seen - `Log (n,2)`:

> Given the case you only have at your disposal certain base measure values, for that your quality property calculation tends to converge to 1 using basic arithmetic operations. Here you might consider to use Log function with basis 2 (please see the derived measure calculation in the example above *Quality Model for MXAM*.)

Given the case that you want to avoid negative values in your measurement function, you can handle this at any time of its calculation using the

- `Abs`-function, that returns the absolute value of the argument.

  Abs([GuidelinesPassedCalc] - [GuidelinesInfoCalc])

or

- `Case... when`: case when ([GuidelinesPassedCalc] - [GuidelinesInfoCalc] < 0) then 1 when ([GuidelinesPassedCalc] - [GuidelinesInfoCalc] > 0) then ([GuidelinesPassedCalc] - [GuidelinesInfoCalc]) else 0 end
- `Power` Returns Arg1 raised to the Arg2 power. Power(Arg1, Arg2)
- `Exp` Another useful application of mathematical functions could be the case of using the `Exp (argument)` function, which returns e raised to the power of the argument. A linear function, such as

  1.0 - ([ModelClones.Clones] / [ModelClones.Subsystems])

  might not be feasible to accurately adapt the Quality Property of ModelClones to your needs. Therefore, you might want to use this function:

  1.0 / Exp(2 * [ModelClones.Clones] / [ModelClones.Subsystems])
- `Ceiling`
- `Floor`
- `Round`
- `Ln`
- `Log`
- `Mod`
- `Power`
- `Product`
- `Sqrt`
- `Min`
- `Max`
- `Avg` This function calculates the average (mean) value of the input parameters: Avg(parameter 1, parameter 2, . . . , parameter n)
- `WeightedAverage`
- `Case... when`
- `IF... else`

12 VALIDATION OF CONFIGURATION

As seen in the previous sections, such as *Project Structures Configuration* or *Quality Model Configuration*, MQC requires the user to take a couple of configuration steps to assure an correct functioning adapted to your data, trends, targets and goals. To point the user to possible configuration errors, MQC is providing the user a validation functionality to guide you through the configuration process of your project.

12.1 VALIDATION OF ARTIFACTS

MQC automatically validates if the imported Artifacts match with your Artifact Structure mentioned in Figure 10.6. In case of lack of congruence, the validation dialogues will inform the user, e.g. about imported but not configured artifacts.

12.2 VALIDATION OF MEASURES

MQC will automatically validate if your imported Base Measures coincide with the ones mentioned in your Quality Model (see *Base Measures and Default Values*) and will give an validation dialogue, e.g. about imported but not configured measures.

12.3 VALIDATION OF CONTEXT CATEGORIES

If you have defined *Context Categories*, like in the example mentioned in see Figure 14.2, MQC will notify you of imported Data Sources, Measurements or Measures, that are imported but were excluded by the user.

13 DATA PROPAGATION

MQC offers a feature to propagate data, which is missing at a certain revision, by using data previously imported instead of importing the data again.

13.1 MOTIVATION

Each piece of data imported into MQC is assigned to a revision depending on the point in time the data was collected, i.e. the report has been created. If this data is not re-imported, MQC treats it as missing for all other revisions.

As long as imported data stays valid and to reduce the amount of missing data, the user may decide to propagate the data, which means to re-use this data in later revisions, too.

With that, MQC is able to calculate and visualize certain quality metrics even for those revisions, where the data used to calculate the particular quality metric has not been re-collected respectively re-imported.

Note: Data may become invalid, if for instance the artifact changes (new model version), but the data is not re-collected.

Therefore the user must decide carefully if and which data to propagate.

In any case, data is replaced if more up-to-date data - new data collected for the same object - is loaded into MQC.

13.2 HOW TO PROPAGATE DATA?

Figure 13.1 shows the availability status of the data currently loaded into MQC. Available data is colored blue, missing data is colored grey. The availability index inside the *Project* KPI shows an availability of only 29 % (for the most recent revision, i.e. CW 44).

To trigger the MQC propagation feature, the user must open the **Settings** dialog from the Configuration section of the menu and select option under the `Propagation` title.

There are three options:

- None (see Section *Remove Propagated Data*)
- All data for the entire project (see Section *Propagate Data For The Entire Project*)

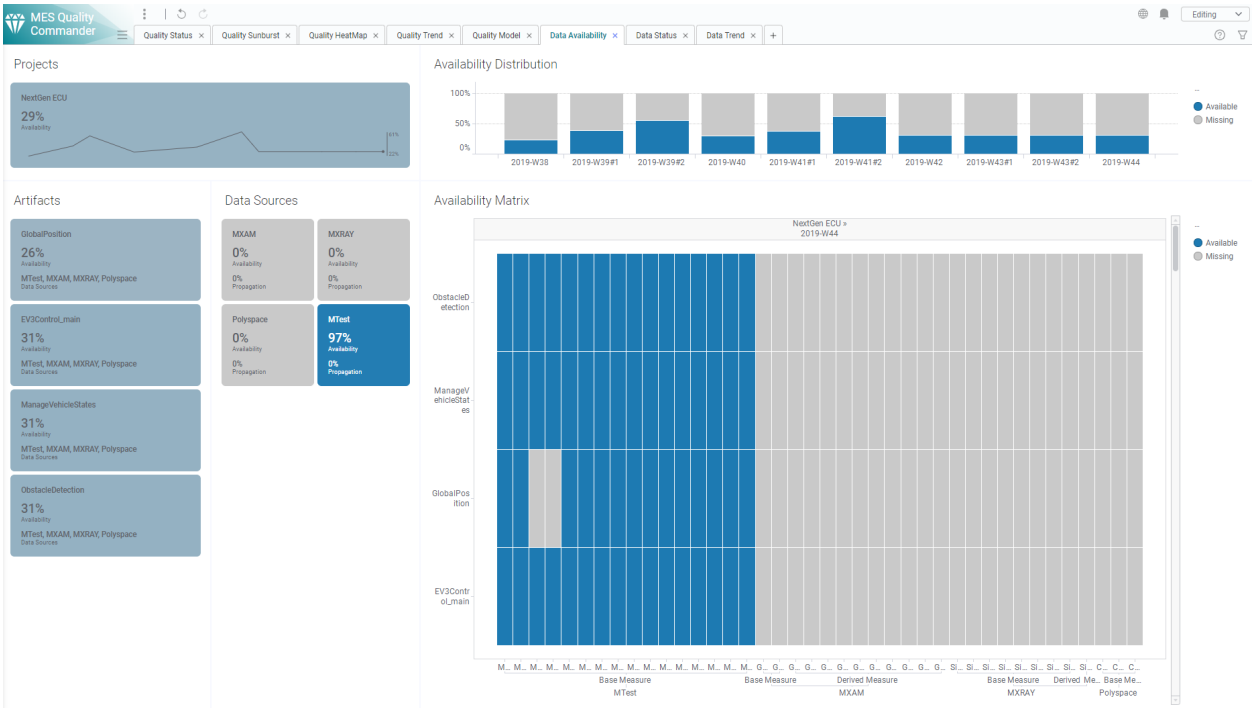


Figure 13.1: Data Availability after Import before Propagation was triggered

- All data towards next milestone (see Section *Propagate Data Towards Next Milestone*)

13.3 PROPAGATE DATA FOR THE ENTIRE PROJECT

Select `All Data` and for **Time range** `For the entire project` under **Propagation** title on **Settings** dialog from the Configuration section of the menu.

This starts with the first revision, for each revision the availability of the data is checked. If data is missing, it is taken from the previous revision if present there (could be either loaded data or again data that was propagated from the revision before).

Figure 13.3 shows the availability status after Propagation has been triggered. Propagated data is colored light blue to distinguish between imported and propagated data. Please note that the availability index inside the *Project* KPI now has been increased to an availability of 99 % and the distribution chart as well as the availability treemap are now showing blocks of propgated data instead of missing data.

Additionally, if at a later point in time new data is loaded for additional revisions, data that is missing for these new revisions is propagated to these revisions, too.

13.4 PROPAGATE DATA TOWARDS NEXT MILESTONE

This use case applies, if replacing missing data from previous revision is only valid for revisions inside one milestone and if for the next milestone new data is expected and should not replaced by data from previous

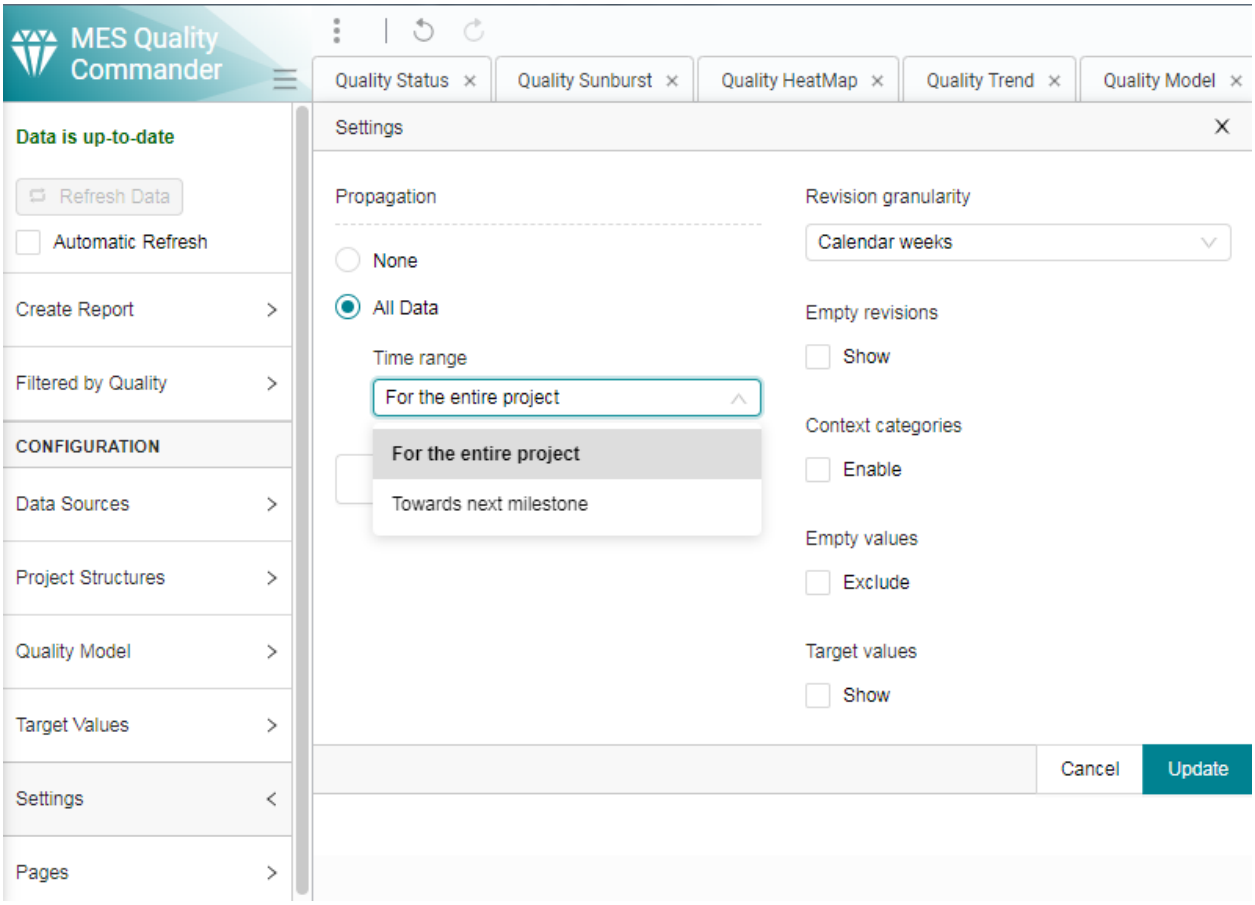


Figure 13.2: Trigger Data Propagation in the Settings dialog

Figure 13.3: Data Availability after Propagation for the entire project

milestone. For more details regarding configuration of project milestones please refer to section *Project Milestone Structure*.

To achieve this, select `All Data` and for **Time range** `Towards next milestone` under **Propagation** title on **Settings** dialog from the Configuration section of the menu.

The mechanism here is the same as *Propagate Data For The Entire Project* but inside (from first to last revision) of each milestone. For each milestone, this fills all revisions up to the last revision of that milestone.

Figure 13.4 shows no propagation for e.g. CW 44, because the corresponding milestone just contains one revision. So, the availability index inside the *Project* KPI stays the same as without propagation. From the other side first milestone contains two revisions (CW 38, CW 39#1) and the distribution chart shows block of propagated data in CW 39#1, where missing data was replaced by data from CW 38.

Note: If propagation was already triggered for entire project, all previously propagated data, which does not match the current selection, will be removed.

13.5 REMOVE PROPAGATED DATA

Select `None` under **Propagation** title on **Settings** dialog from the Configuration section of the menu.

This removes all propagated data from all revisions.

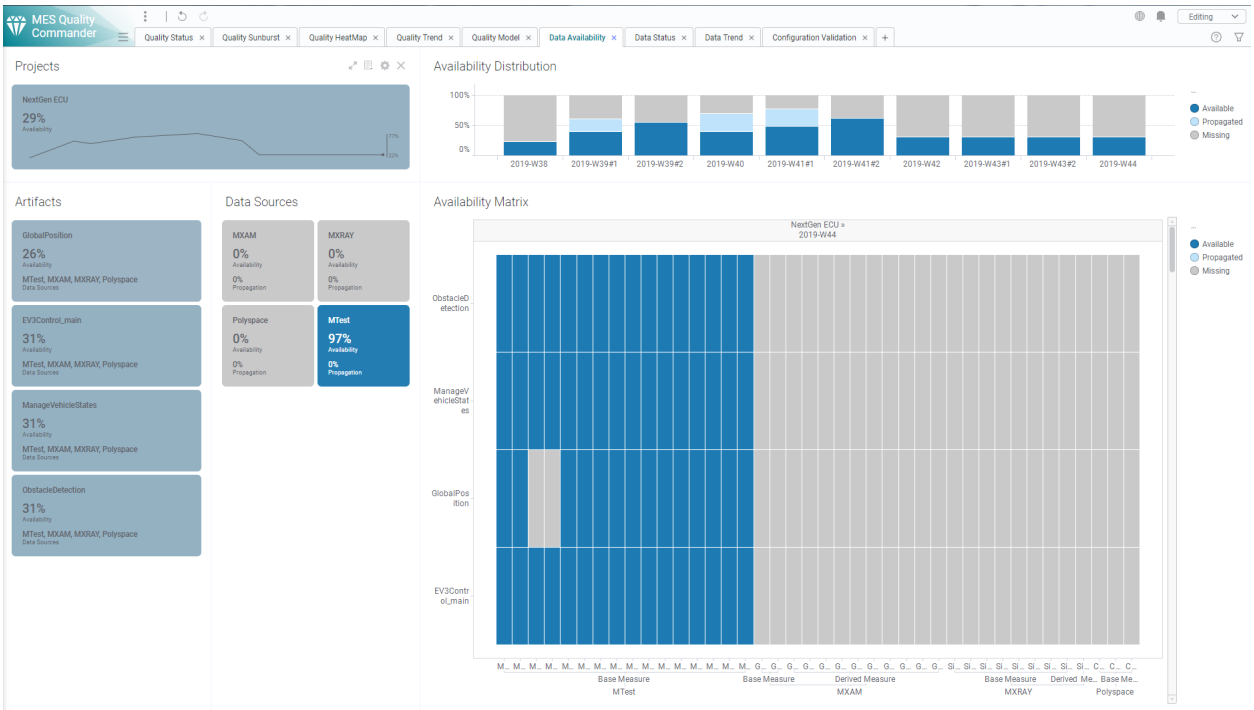


Figure 13.4: Data Availability after Propagation towards next milestone

14 CONTEXT CATEGORIES

When defining your Project Milestone Structure in MQC (see *Project Milestone Structure*), by default each measure imported for a particular artifact for a certain revision is expected for all other artifacts and revisions, too. This may produce lots of missing data, which might not always be what we want.

By using context categories you can assign data sources to artifacts, thus defining, which data (from which data sources) is expected for a certain artifact. In that way, data previously stated as "missing" - and probably resulting in bad quality - is treated as "excluded" data. This means it is ignored in availability and quality calculation.

There are two steps required to create your Context Categories: First, specify them within the project structures (*Define your Context Categories*) and edit them in the Quality model (*Add whitelists/ blacklists to your Context Categories*).

14.1 DEFINE YOUR CONTEXT CATEGORIES

The "Artifact Structure" sheet of the Project Structures Excel file (see *Project Structures Configuration*) contains an additional column *ContextCategories* to specify your own Context Categories (see Figure 14.1).

	A	B	C	D	E	F
1	ArtifactName	ContextCategories	ArtifactWeight	StructureRoot	StructureGroup	StructureElement
2	EV3Control_main	EV3_CC	1		Model	Main
3	ObstacleDetection		1		Model	Function
4	GlobalPosition	Global_CC	1		Model	Function
5	ManageVehicleStates		1		Model	Function
6						

ProjectArtifactStructure | ProjectMilestoneStructure | **ArtifactStructure** | ArtifactMapping

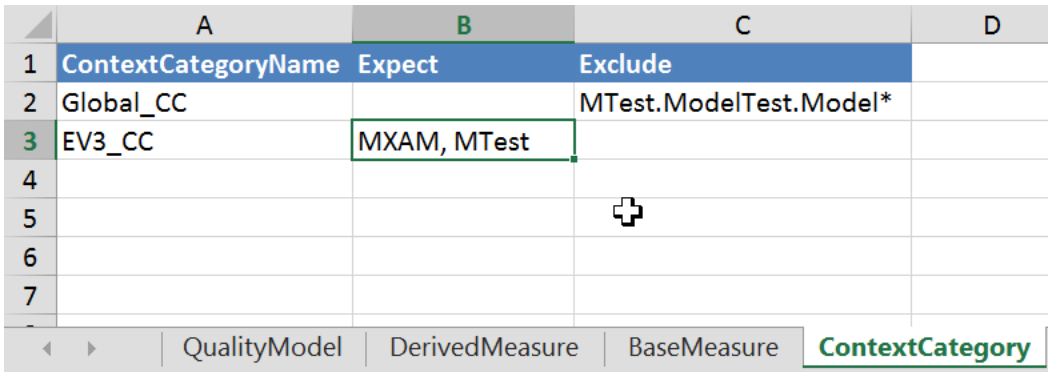
Figure 14.1: Specify your ContextCategoryName here. You can chose any name you'd like.

Note: You can assign any name you'd like, except the string "none" to exclude artifacts completely from your project.

Note: When no Context Category is defined, i.e. the field is kept empty, all available data is shown.

14.2 ADD WHITELISTS/ BLACKLISTS TO YOUR CONTEXT CATEGORIES

With MQC you are able to edit your Context Categories by using whitelists and blacklists, respectively. Within the *ContextCategories* sheet of the quality model (see *Quality Model Configuration*) you can assign to your specified Context Category names a comma-separated list of data source names, Measurements, Measures or certain strings of expected data or data that should be excluded. The following figure gives one example of blacklist to exclude all measures from the MTest measure ModelTest starting with the string Model.. and another example of using the whitelist to only consider certain Data Sources (here MXAM and MTest) (see Figure 14.2).



	A	B	C	D
1	ContextCategoryName	Expect	Exclude	
2	Global_CC		MTest.ModelTest.Model*	
3	EV3_CC	MXAM, MTest		
4				
5				
6				
7				

QualityModel | DerivedMeasure | BaseMeasure | ContextCategory

Figure 14.2: Editing context categories. For each Context Category, a comma-separated list of Data Sources, Measurements or Measures can be configured.

For example, you can expect for a ContextCategoryName one or more Data Sources resp. tool names and exclude only certain Measures, using the following sytax:

- `MTest.*.*Requirements*, MTest.*.*Total*`

Here all MTest measures that contain the string Requirements or Total are excluded from the MQC project.

Please note, configured context categories will *NOT* be applied automatically after loading a Project Structures and quality model file (see Figure 14.3).

To use the defined context categories, enable the **Context Categories** check box in the Settings dialog from the Configuration section of the menu **Settings** as shown in Figure 14.4).

Now, for each artifact only data from those expected data sources is shown in all visualizations as well as only expected data is used for calculating quality. By this, you can easily distinguish between not expected data and data that is really missing.

On enabling the check box, MQC provides a notification if data was imported that is now excluded by means of context categories. For each artifact you will get a list of not expected data sources. On one hand, this shows surplus data, but it may also indicate an incomplete context category configuration.

Disabling the check box keeps the imported configuration, whereas it is no longer used. Hence, all visualizations and calculation are updated accordingly.

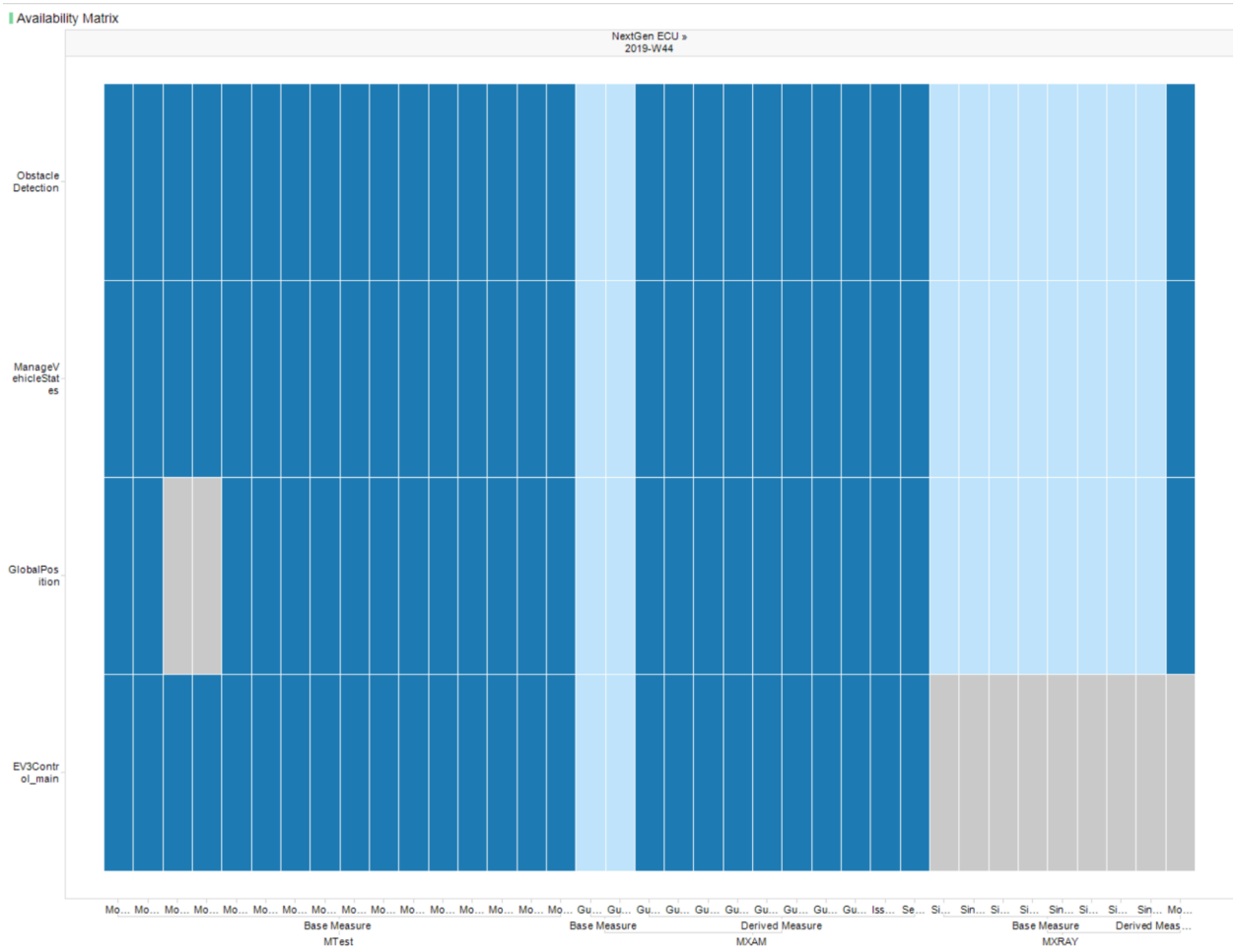


Figure 14.3: Availability matrix with all data (yet not activated Context Categories)

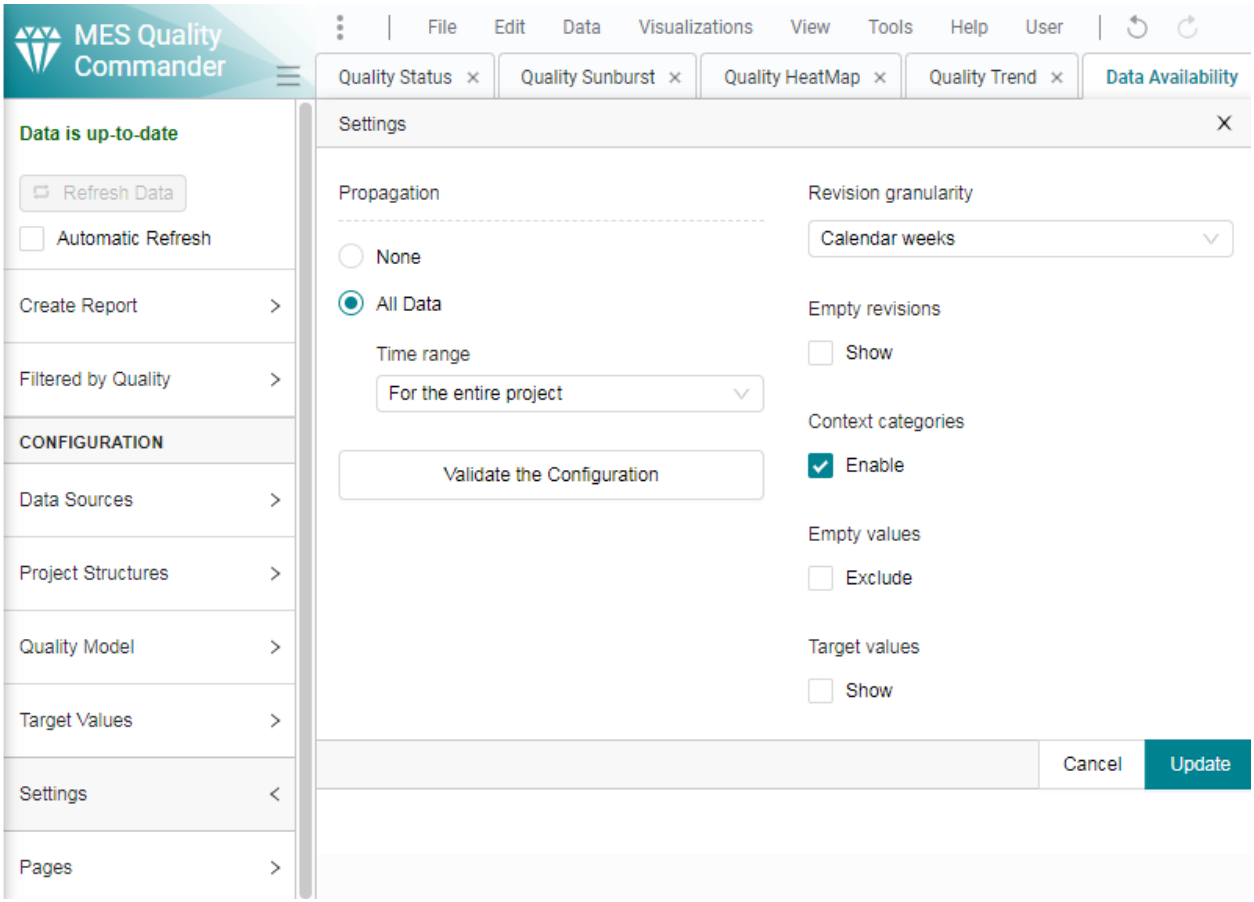


Figure 14.4: Click on the check box to activate Context Categories

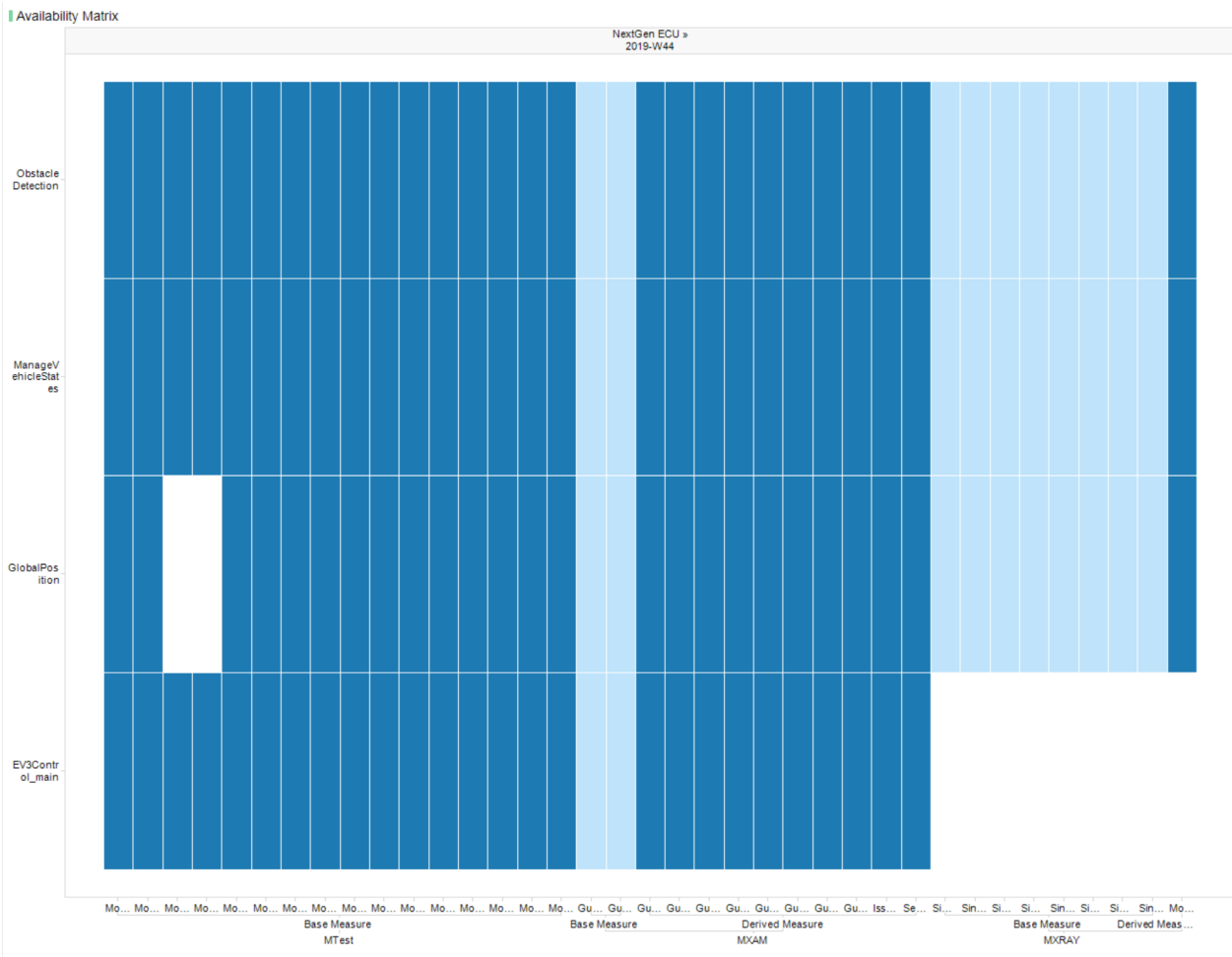


Figure 14.5: Availability matrix just showing expected data and white areas for excluded data.

15 TARGET VALUES

MQC provides the possibility to add target values to selected measures, i.e. base measures as well as quality properties, to enable the user to compare the current state of a project and its reported data against an expected state.

The User is able to define:

- target values for each artifact;
- target values for each milestone (whereas target values can be defined for selected milestones only);
- one or multiple target values per measure.

MQC allows:

- to define arbitrary target names (target names for the same measure must be unique);
- to import a target value configuration at any time replacing previously imported targets;
- to completely remove imported target values;
- to switch on or off the visualization of target values.

MQC shows target values in trend charts as well as in status charts of custom pages (see *Custom Pages*).

In trend charts target values are shown as additional dashed line per measure trend line using the same color as used for the measure itself (see Figure 15.1).

In status charts target values are shown as horizontal line per measure bar (see Figure 15.2).

15.1 CONFIGURATION OF TARGET VALUES

Target values have to be configured for milestones, hence, targets are intended to be reached at the end of the corresponding milestone. Based on the configured milestone targets, MQC calculates target values for revisions (see details in *Calculation of Target Values per Revision*).

The configuration of target values in MQC is done via Excel and must have the following structure:

- **Artifact Name:** the name of the artifact a target value for a certain measure shall be applied to
- **Milestone Name:** the name of the milestone a target value for a certain measure shall be reached
- **Target for Measure Name:** for each measure, a target value shall be applied to, a separate column has to be filled; the column header must contain the full qualified name of the respective measure and is expected in the following notation:

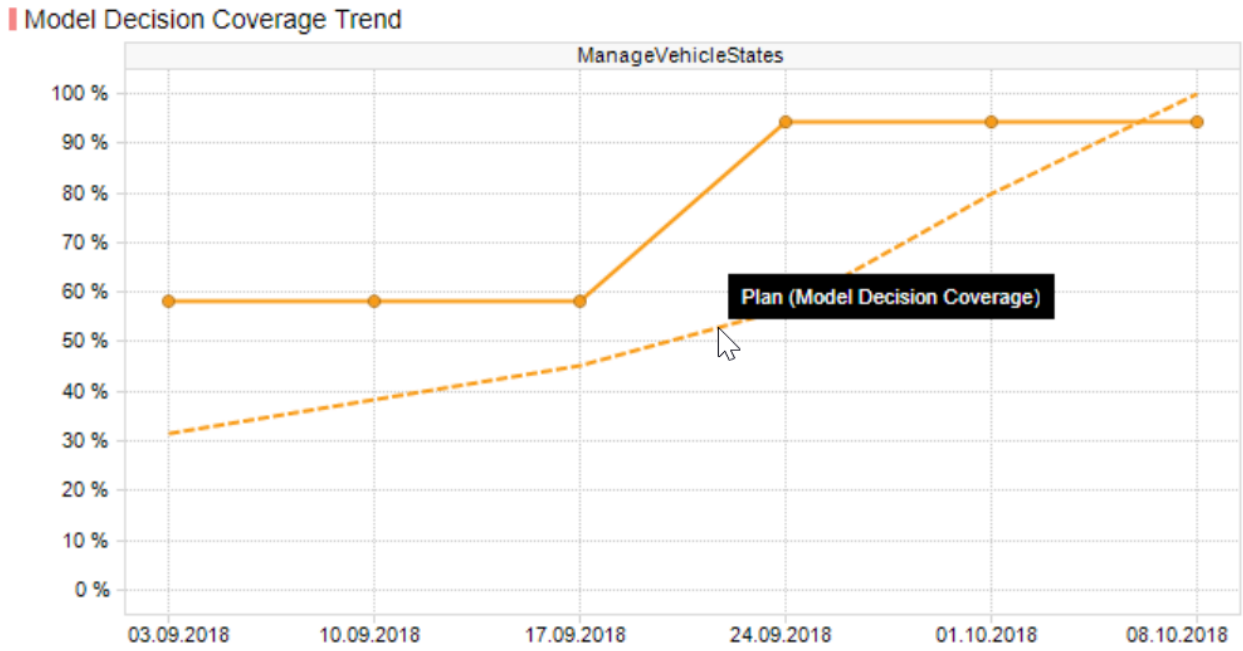


Figure 15.1: Trend chart with one measure incl. target trend

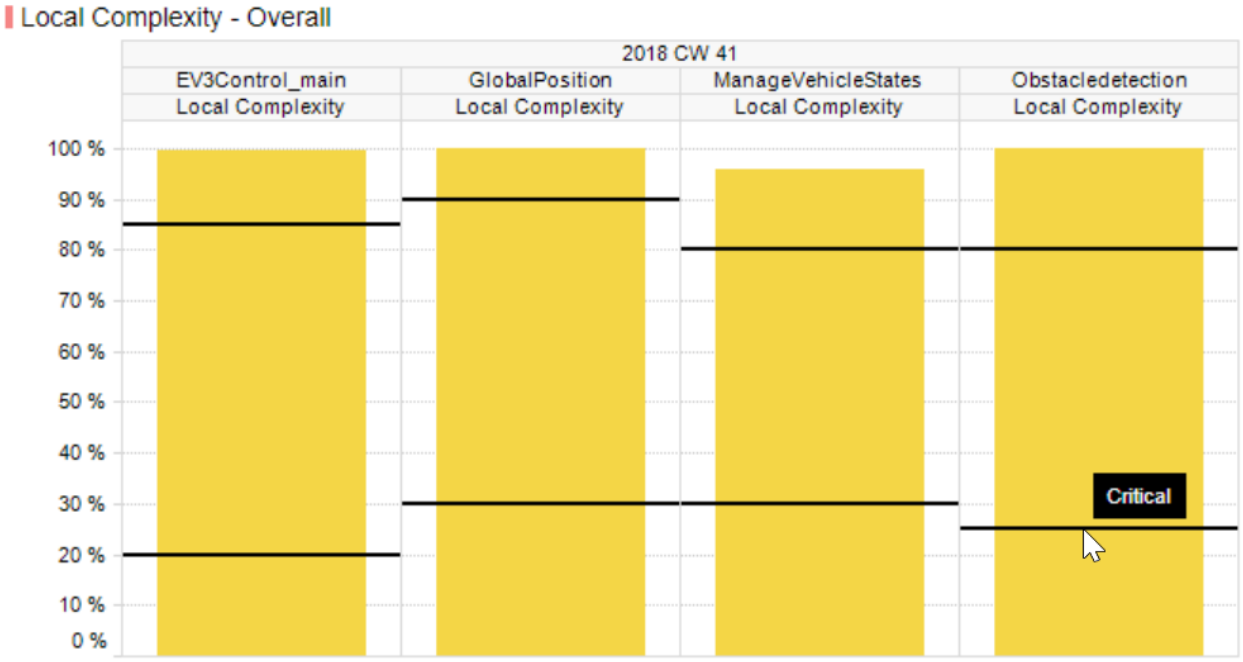


Figure 15.2: Status chart with one measure incl. two target lines

- Targets for base measures (as defined in the Quality Model Excel file `TargetValuesPerMilestone_Template.xlsx` to be found under: `C:\Users\...\AppData\Local\TIBCO\Spotfire\10.3.3\Modules\MES.MQC.CoreExtension_X.X.X.X\Resources\SampleFiles\` within the "Base Measure" sheet): **DataSourceName.MeasurementName.BaseMeasureName.VariableName.TargetName**
- Targets for quality properties (as defined in the Quality Model Excel file within the "Quality Model" sheet): **QualityPropertyName.TargetName**

	A	B	C	D	E
1	ArtifactName	MilestoneName	Model Decision Coverage.Plan	Local Complexity.Critical	Local Complexity.Acceptable
2	ManageVehicleStates	Static Analysis Start		30	80
3	ManageVehicleStates	Module Test Start	50	30	80
4	ManageVehicleStates	Static Analysis Full		30	80
5	ManageVehicleStates	Model Test	100	30	80
6	ManageVehicleStates	This is the Day		30	80
7					
8	EV3Control_main	Static Analysis Start		20	85
9	EV3Control_main	This is the Day		20	85
10					
11	GlobalPosition	Static Analysis Start		30	90
12	GlobalPosition	This is the Day		30	90
13					
14	Obstacledetection	Static Analysis Start		25	80
15	Obstacledetection	This is the Day		25	80
16					

Target Values

Figure 15.3: Several options to configure target values per measure, per artifact and per milestone using different target names

Figure 15.3 shows different options to configure target values per quality property:

- for selected milestones (not the first and not the last one),
- for all milestones,
- for the first and the last milestone.

Please note, if the configured measure name does not contain a target name at the end, a default target name **"Target"** is applied by MQC instead.

To import a target value configuration Excel file, open the Target Values dialog from the Configuration section of the menu, click on **Target Values/Load** and navigate to the folder, where you have stored your configuration.

Importing a new target value configuration will replace all previously imported target values.

To make the imported target values visible in the corresponding visualizations on custom pages, enable the **Show Target Values** check box in the Settings dialog from the Configuration section of the menu.

Disabling the check box just removes the targets from the visualizations, whereas the target value configuration still remains in MQC.

If the **Show Target Values** check box is already enabled when importing a target value configuration, MQC will automatically update all corresponding visualizations. The same applies, when custom pages are added or replaced. MQC will automatically include already imported target values into the custom pages visualizations.

To remove all configured target values, choose **Target Values/Remove** in the Target Values dialog from the Configuration section of the menu. MQC then automatically updates all corresponding visualizations.

15.2 CALCULATION OF TARGET VALUES PER REVISION

Via configuration, target values can be assigned to arbitrary milestones, i.e.

- to all configured milestones (see *Project Milestone Structure*)
- to one or more but not all milestones

MQC supports target value gaps (one or more milestones without targets)

- at the beginning of a project
- in the mid of a project
- as well as at the end of a project.

Target values are defined per (end of) milestone. Measures and quality properties in MQC are always handled per revision. Thus, MQC calculates target values per revision based on the configured milestone target values.

MQC interpolates between previous and next configured milestone target value. As init value, which is assigned to the project start date, MQC uses 0.

After the last milestone with configured target value, MQC keeps the last target value. Interpolation currently is done linearly.

Note: For a comprehensive target value calculation, a proper milestone configuration is needed. Each milestone must contain a valid milestone start and due date. Please refer to *Project Milestone Structure*.

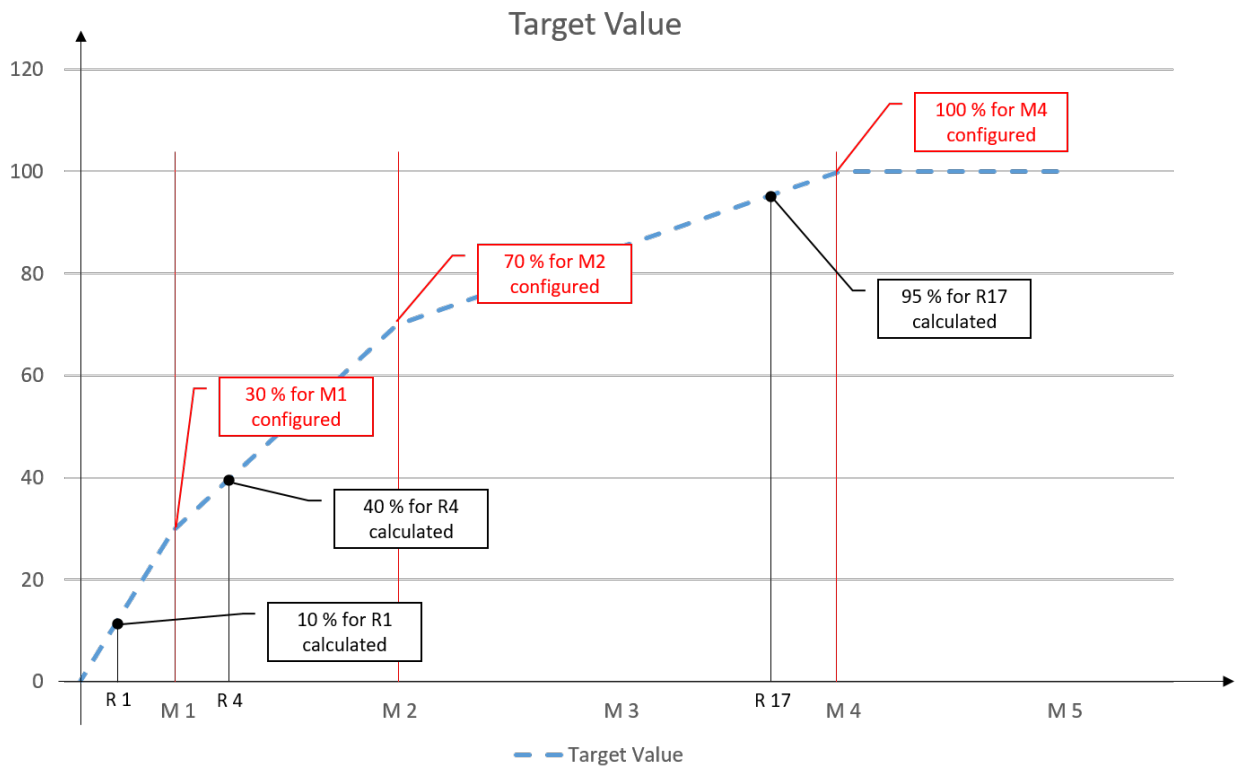


Figure 15.4: Target values configured for milestones M1, M2 and M4, linearly calculated target values for revisions assigned to milestones

16 CUSTOM PAGES

MQC provides the possibility to create user-defined trend and status charts for selected measures, i.e. base measures, derived measures, as well as quality properties, to better structure the data to be analyzed. These additional visualizations are shown on extra pages created on demand, whenever a new configuration of such charts is imported into MQC.

The User is able to define:

- multiple custom pages
- multiple visualization charts per custom page
- multiple measures per chart

MQC allows:

- to import a custom page configuration at any time and with that to replace a previously imported configuration
- to add custom pages to the current analysis by importing a new custom page configuration
- to update already created custom pages by replacing the custom page configuration

Note, that imported target values will be shown on custom pages only. This applies to all those targets related to measures configured by the user to be shown in a custom page visualization. Showing targets for measures can be switched on or off. Refer to *Target Values* for more details.

16.1 CUSTOM PAGE STRUCTURE

All custom pages follow the MQC standard layout of the MQC data pages:

- a Project KPI chart on the top-left showing current data availability
- an Availability Bin Distribution chart on the top-right
- an Artifact KPI chart on the bottom-left showing all artifacts in alphabetical order
- the configured visualizations are shown in the main area of the page.

The following picture shows an example for a user-defined page with two status charts. For the configuration refer to Figure 16.5.

Status values for all or selected artifacts for one revision are shown in a bar chart.

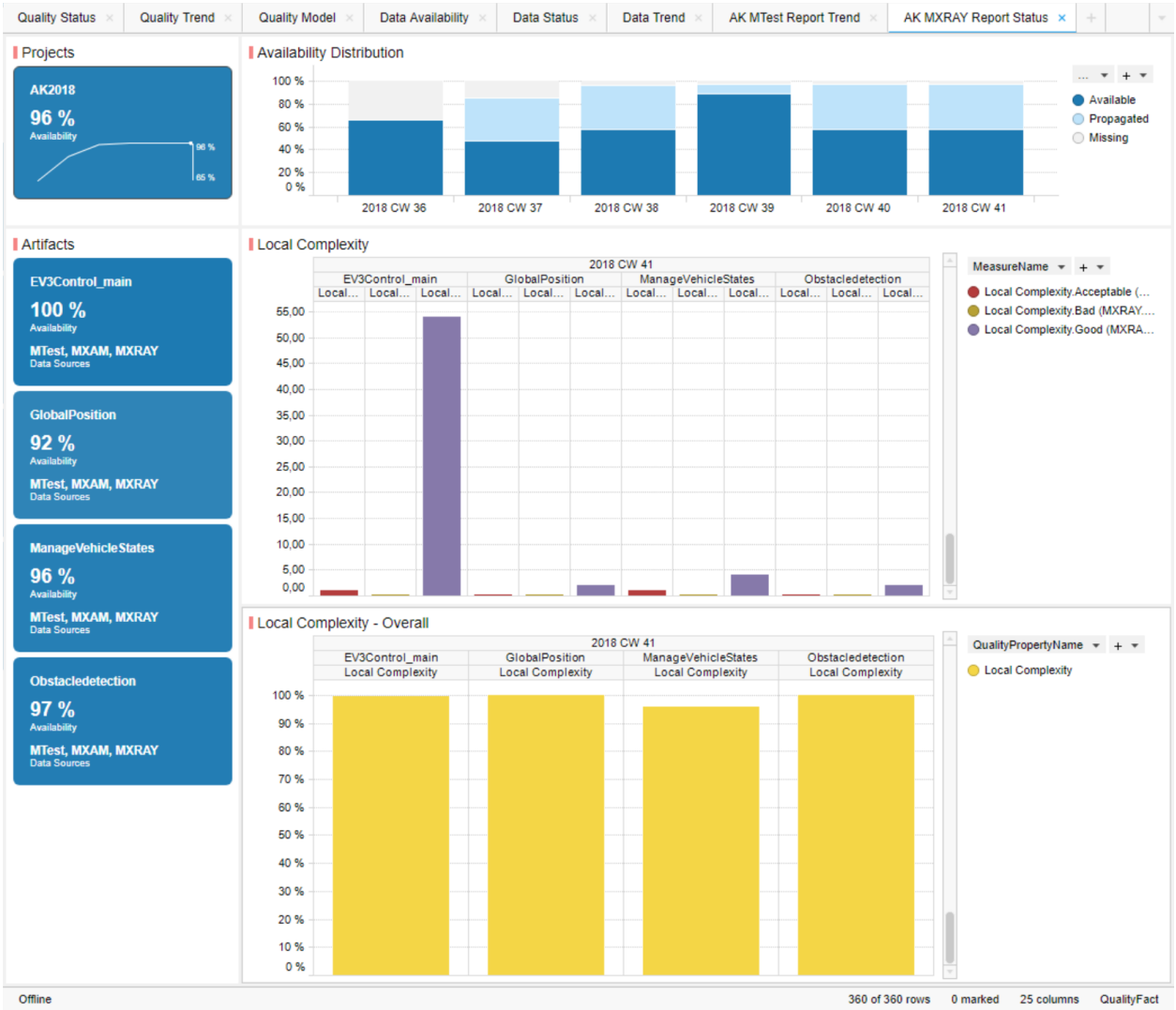


Figure 16.1: Custom Page showing Status Charts for selected Base Measures and Quality Property

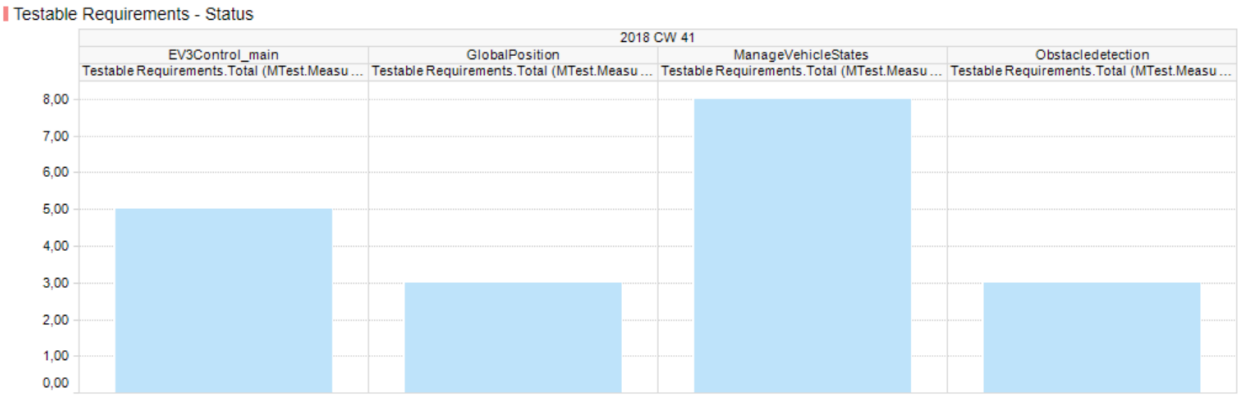


Figure 16.2: Status chart showing one status value per artifact

If multiple measures are configured to be visualized within the same chart, these measures will be shown as side by side bars per artifact.

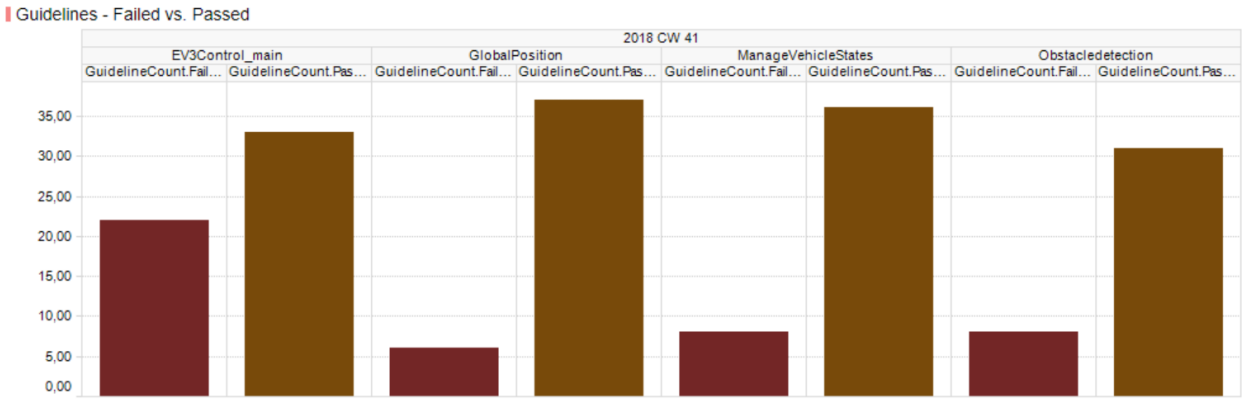


Figure 16.3: Status chart showing multiple status values per artifact

Multiple revisions are trellised, i.e. the chart always shows the measure status for one revision at a time with the possibility to scroll between revisions (see Figure 16.1).

Corresponding target values are shown as horizontal line per measure bar.

The trend over time for selected measures is shown within a line chart. Each measure trend is always shown as line over multiple revisions or as a dot if just one revision is marked by the user.

Multiple artifacts are trellised, i.e. the chart always shows the measure trend lines for one artifact at a time with the possibility to scroll between different artifacts.

If multiple measures are configured to be visualized within the same chart, one line per measure using different colors is shown at the same time for the current artifact.

Corresponding target values are shown as additional dashed line using the same color as for the measure.

The following picture shows an example for a user-defined custom page with two status charts. For the configuration see Figure 16.5.

16.2 CUSTOM PAGE CONFIGURATION

Custom pages and charts are configured using Excel.

The configuration has the following structure:

- **Page Title:** a user defined title for the current custom page
- **Chart Title:** a user defined title for the current chart to be shown within the current custom page
- **Chart Type:** the type of the current chart, which must be either “Status” (for status bar charts) or “Trend” (for trend line charts)
- **Measure Name:** the full qualified name of the measure to be shown in the current chart; measure names are expected in the following notation:
  - Base Measures: **DataSourceName.MeasurementName.BaseMeasureName.VariableName**

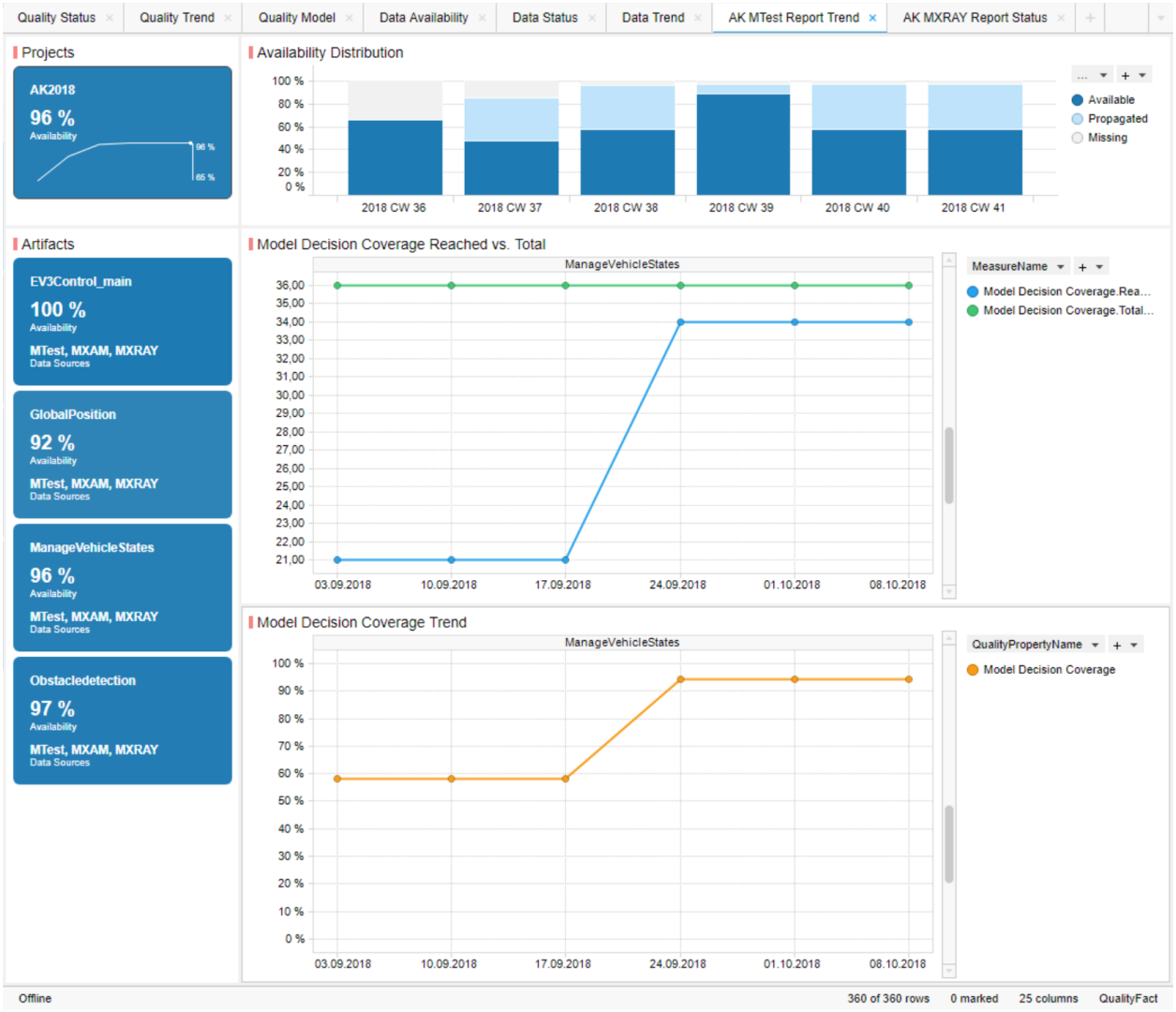


Figure 16.4: Custom Page showing Trend Charts for selected Base Measures and Quality Property

	A	B	C	D	E	F
1	**Page Title**	**Chart Title**	**Chart Type**	**Measure Name**	**Chart Data Reference**	
2	AK MTest Report Trend	Model Decision Coverage Reached vs. Total	Trend	MTest.Measurement.Model Decision Coverage.Reached	Data	
3	AK MTest Report Trend	Model Decision Coverage Reached vs. Total	Trend	MTest.Measurement.Model Decision Coverage.Total	Data	
4	AK MTest Report Trend	Model Decision Coverage Trend	Trend	Model Decision Coverage	Quality	
5	AK MXRAY Report Status	Local Complexity	Status	MXRAY.SingleSystemResults.Local Complexity.Good	Data	
6	AK MXRAY Report Status	Local Complexity	Status	MXRAY.SingleSystemResults.Local Complexity.Acceptable	Data	
7	AK MXRAY Report Status	Local Complexity	Status	MXRAY.SingleSystemResults.Local Complexity.Bad	Data	
8	AK MXRAY Report Status	Local Complexity - Overall	Status	Local Complexity	Quality	
9						

Report Pages Configuration

Figure 16.5: Configuration of two additional pages, each page with two charts

  - Derived Measures: **DataSourceName.DerivedMeasureName**
  - Quality Properties: **QualityPropertyName**

- **Chart Data Reference:** the type of the current measure to be shown in the current chart, which must be either "Data" (for base measures and derived measures) or "Quality" (for quality properties)

Note: It is not possible to combine different types of measures within the same chart! A chart must either contain data measures, i.e. base measures and/or derived measures, or quality measures, i.e. quality properties.

It is possible to combine different chart types (status and trend) on the same page, whereas it is not recommended. This is because of the specific marking behaviour, which is not the same for status and trend charts. Selecting a specific artifact and/or a specific revision affects all visualizations of a certain page.

16.3 CONFIGURATION DATA IMPORT AND PAGE CREATION

To import a custom page configuration Excel file `CustomPagesConfig_Template.xlsx` (to be found under: `C:\Users...\AppData\Local\TIBCO\Spotfire\10.3.3\Modules\MES.MQC.CoreExtension_X.X.X.X\Resources\SampleFiles\`), in the Pages dialog from the Configuration section of the menu press 'Load' button and navigate to the folder, where you have stored your configuration.

Importing a new configuration Excel file will replace any previously imported custom page configuration.

After import, MQC will:

- automatically create a new page for each configured custom page including all configured visualizations, if a custom page with the corresponding title does not yet exist
- automatically replace all existing pages, where the title matches a configured custom page, hence, all page visualizations of a replaced page are updated according to the newly imported configuration

Note, that MQC will not close any custom page defined by a previously imported configuration, if the current configuration file does not contain a page with an identical title. Those pages have to be manually closed by the user.

Also note, that MQC will not replace any MQC Standard or Tool Page, if the configuration file contains a page with a title that matches the title of an MQC Standard or Tool Page. Instead an additional page with that name is created. Nevertheless, it is highly recommended to use dedicated names within the custom page configuration file.

17 DATA DETAILS

In MQC there are three ways to retrieve detailed information from the imported data:

- *Data Details Drill Down* from quality measures to the underlying data that was used to compute the quality,
- *Tool Pages* to get detailed information on tool specific data,
- *Custom Pages* to create project-specific pages for selected measures (quality as well as data) including targets.

17.1 DATA DETAILS DRILL DOWN

The data details drill down functionality provided by MQC is the easiest way to track the underlying data, which was used to calculate a certain quality measure value as shown on the MQC Quality pages.

The drill down works by selecting the data of interest by either marking in the charts on a Quality page or by using the "Data of the Quality" / "Filter by Quality" dialog in the left menu.

When on a Quality page clicking inside a chart results in a marking, which is then used to reduce the shown quality in other charts depending on the selection.

For the underlying data of quality, mark either

- an arbitrary number of Quality Properties
- an arbitrary number of Quality Bins in the Quality Bin Distribution chart
- an arbitrary number of quality tiles within the *QualityMatrix* visualization on the Quality Status page or within the *Quality Status* on the Quality Sunburst page

---

Note: The way how data is selected (KPIs or tiles) defines the data that is shown at the Data pages, see *Data Selection for Drill Down*.

After selecting the quality measures of interest, the menu item "Data of the Quality" in the left menu get's prepended with a grey link button.

By clicking on the grey link button opens the corresponding data page to the currently open quality page (e.g. Data Status for Quality Status) and reduces the metrics shown in the main visualizations to those used to calculate the selected quality measures.

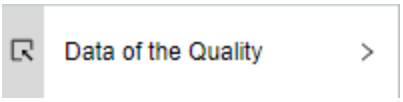


Figure 17.1: "Data of the Quality" in left menu, when something is marked

Clicking on the menu item "Data of the Quality" itself opens the "Data underlying the Quality" dialog, where the current marking is applied to the presented checkboxes. Here the selection can be changed indepentant from the marking and afterwards applied for the data drill down with "Filter data"

17.1.1 Data Selection for Drill Down

Quality Bin Distribution

If one or more Quality Measure Bins are selected, MQC shows the data, which resulted in the selected quality bins.

Quality Properties

If one or more Quality Properties are selected, MQC shows the data, which was used to calculate the selected quality properties.

Quality Tile Selection

Each quality tile inside the *Quality Matrix* on the Quality Status page shows the quality measure for a particular quality property for a particular artifact.

The markable parts of the *Quality Status* on the Quality Sunburst page show Quality Properties and grouped Characteristics.

If one or multiple tiles are selected, MQC only shows the data, which was used to calculate exactly the selected quality property for the selected artifact.

17.1.2 Drill down to Data Trend

To drill down to data trend, move to the Quality Trend page, and select the data of interest, e.g. all acceptable (yellow) and bad (red) quality bins from the *Quality Bin Distribution* chart at the top of the page.

The other visualizations immediately reduces the shown quality properties to those with an acceptable or insufficient quality per artifact. By hovering over the tiles the concrete quality value is shown.

Now click on the grey button next to "Data of the Quality" in the left menu.

MQC directly switches to the Data Trend page and reduces the visualization to the measures that were used to calculate the selected quality (see Figure 17.4).

17.1.3 Drill down to Data Status

Instead of drilling down to data trend, MQC also supports to drill down to data status.

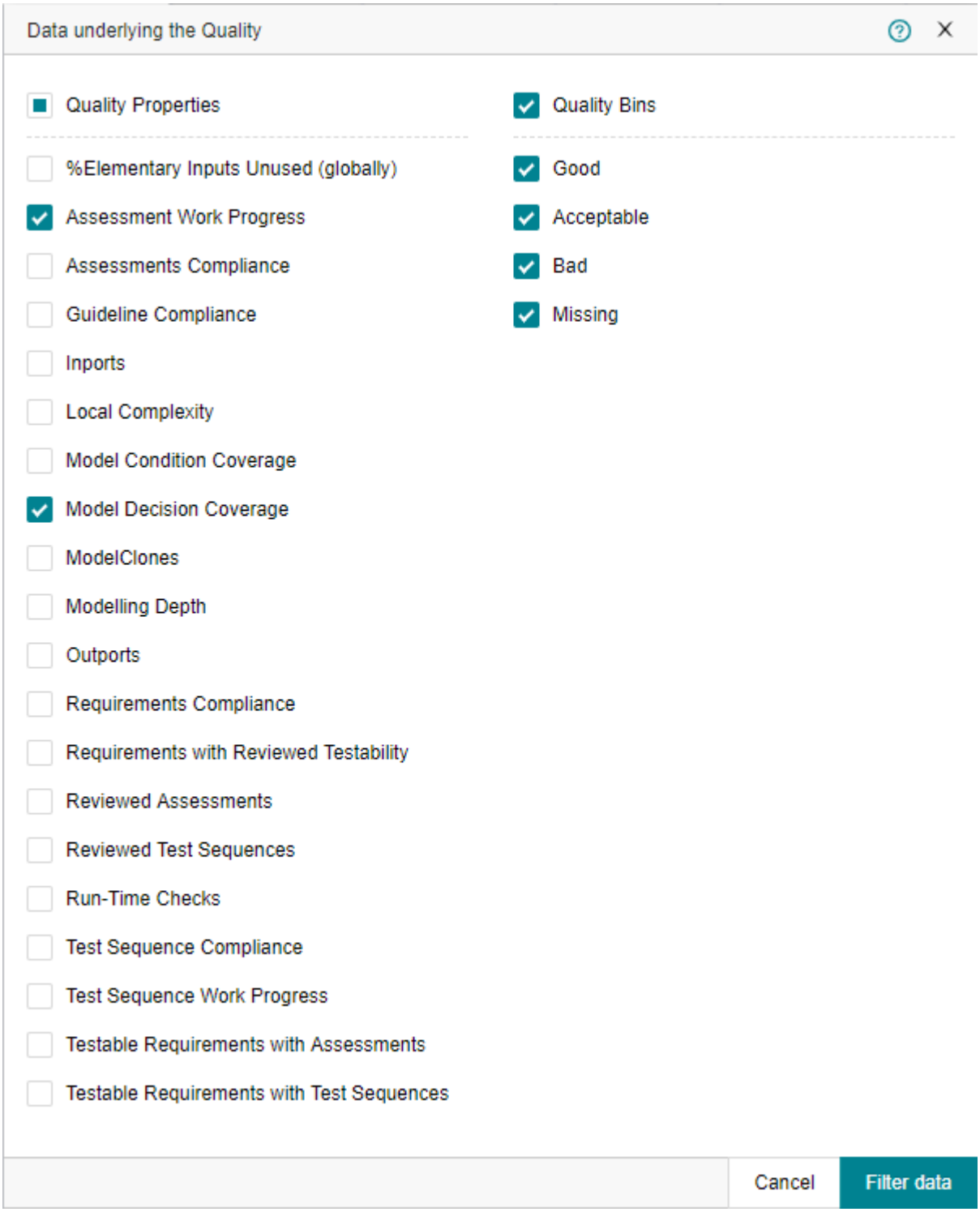


Figure 17.2: "Data underlying the Quality" dialog with only some Quality Properties and all Quality Bins checked

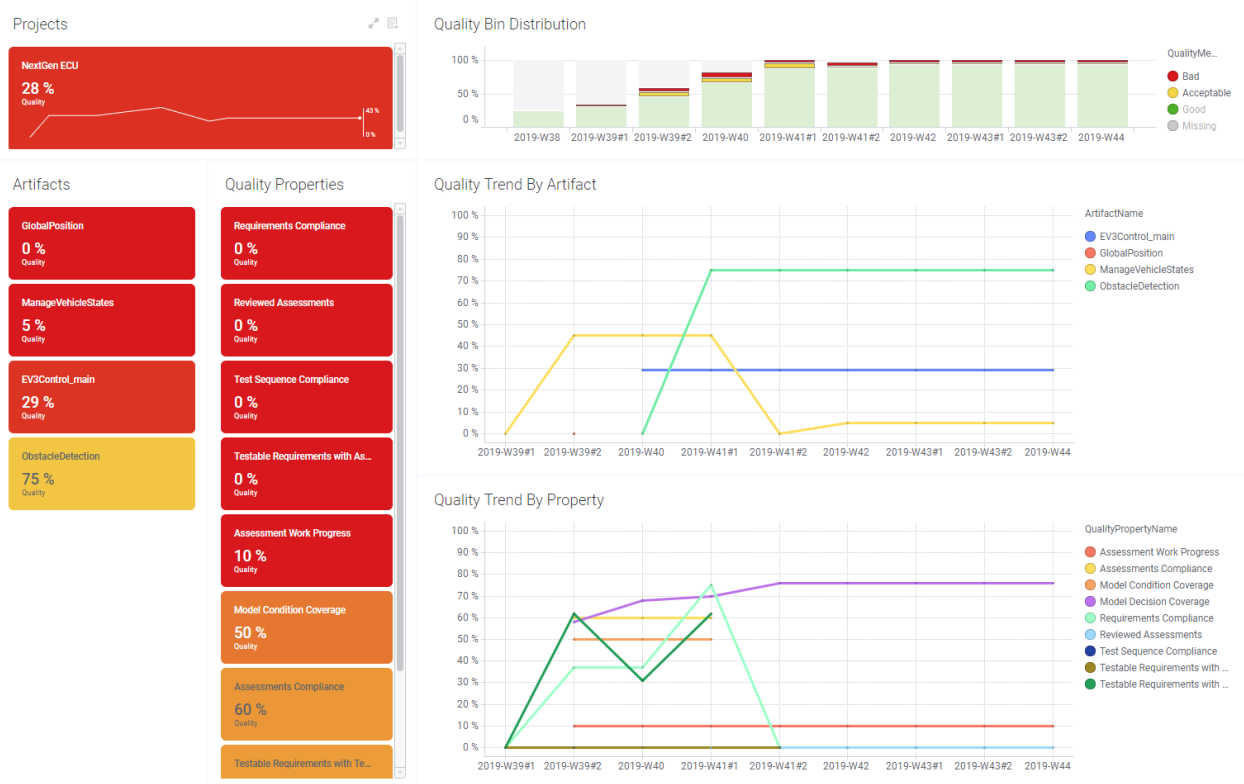


Figure 17.3: MQC Quality Trend page after marking acceptable and bad quality bins to reduce the main visualization to quality properties of interest

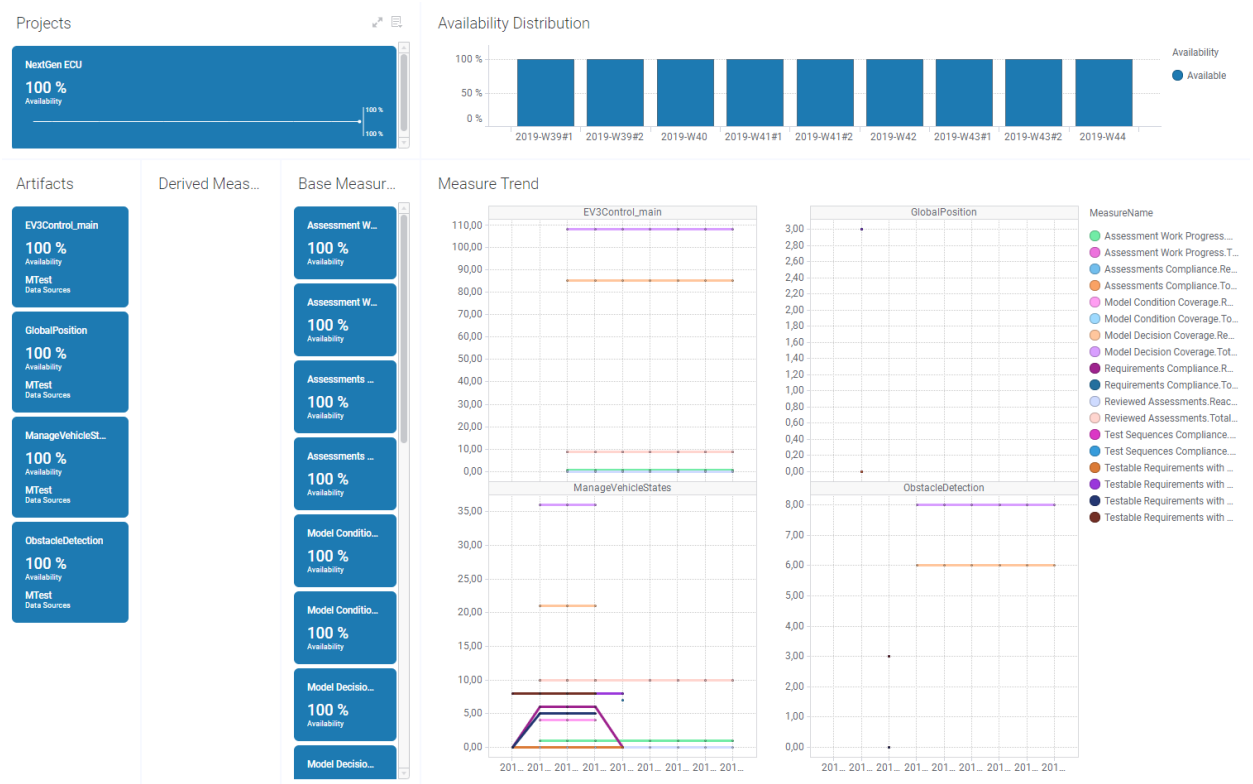


Figure 17.4: MQC Data Details on Data Trend showing underlying data

Switch to a Quality Status page (Quality Status, Quality Sunburst or Quality Heatmap), and select multiple artifacts as well as multiple quality properties.

The main visualization are limited based on your selection, so, the *Quality Matrix* visualization just shows qualities for selected artifacts and selected quality properties.

Now click on the grey button next to “Data of the Quality” in the left menu.

MQC directly switches to the Data Status page and reduces the visualization to those base and derived measures that were used to calculate each combination of the selected artifacts and quality properties (see Figure 17.6).

17.2 TOOL PAGES

A second way to look at the imported data is by using preconfigured tool pages. MQC provides the possibility to add specific tool pages for many of the supported data source adapters on demand.

17.2.1 MXAM Details Page

The MXAM Details page can be added via checked `MXAM` under the *Tool Pages* title in **Settings** dialog from the Configuration section of the menu or by **Tools/MQC Tool Pages/Add MXAM Details Page**.

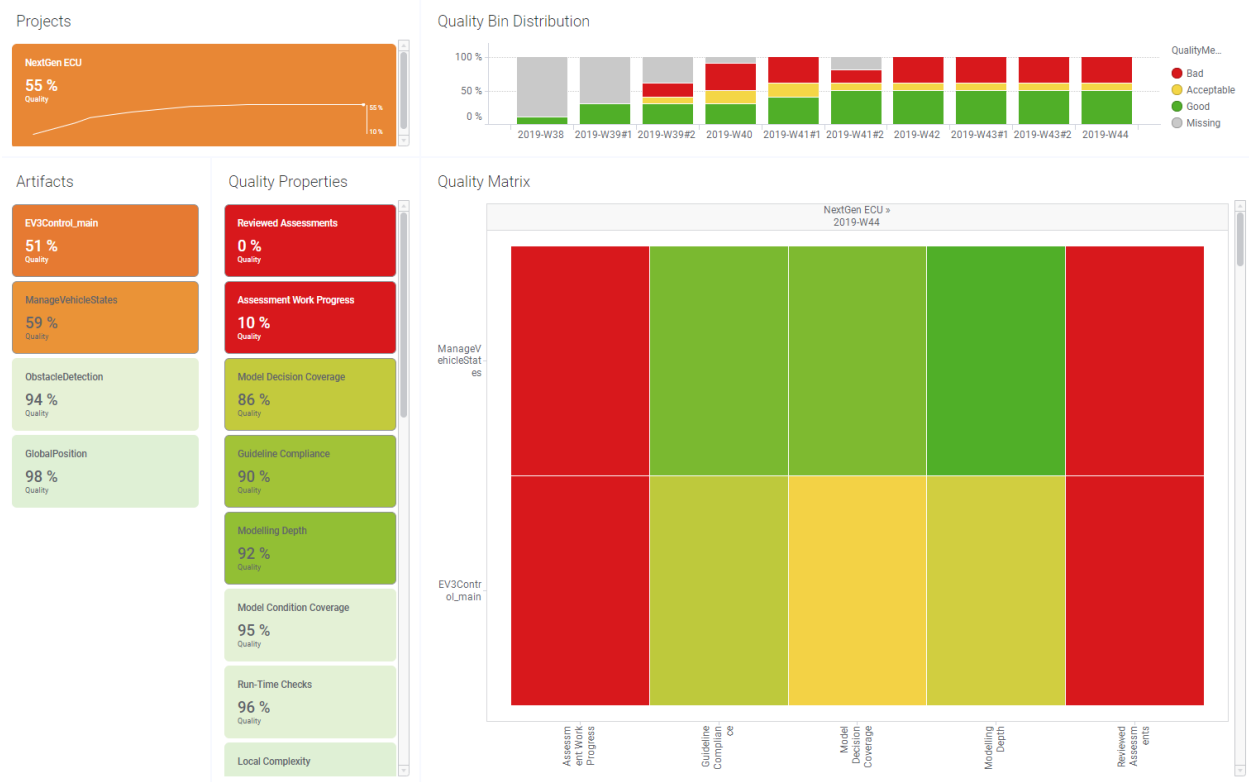


Figure 17.5: MQC Quality Status page just showing selected Artifacts and Quality Properties

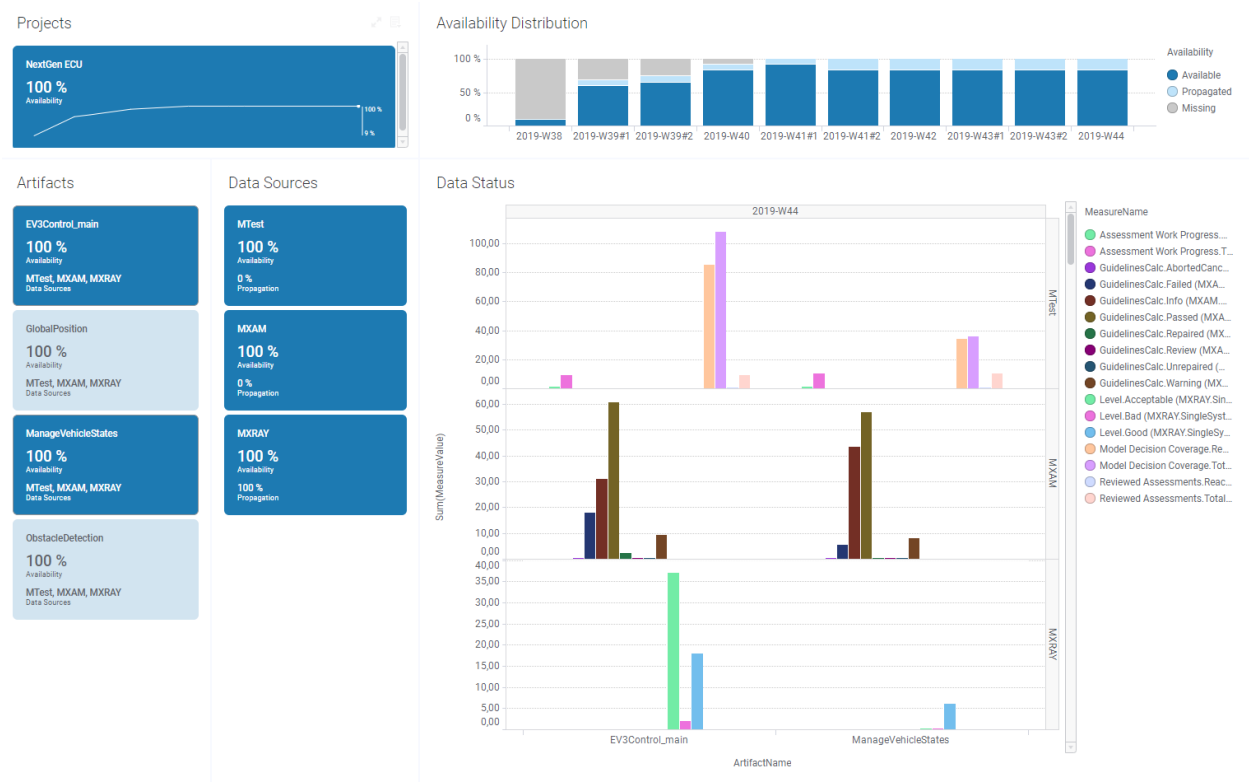


Figure 17.6: MQC Data Details on Data Status showing underlying data

The MXAM page focuses on MXAM findings per artifact, additionally showing number and type of severe issues to be solved first.

For more details, please refer to Section *MXAM Tool Page*.

17.2.2 MTest Details Page

The MTest Details page can be added via checked `MTest` under the *Tool Pages* title in **Settings** dialog of the Configuration panel or by **Tools/MQC Tool Pages/Add MTest Details Page**.

It provides a comparision of expected versus reached test metrics in trend and in status, which is structured based on:

- Requirements
- Test Sequences
- Assessments
- Coverage

For more details, please refer to Section *MTest Tool Page*.

17.2.3 M-XRAY Details Page

The M-XRAY Details page can be added via checked `MXRAY` under the *Tool Pages* title in **Settings** dialog of the Configuration panel or by **Tools/MQC Tool Pages/Add MXRAY Details Page**.

The M-XRAY page focuses on M-XRAY results per artifact, e.g. how many subsystems per artifact with good, acceptable or bad local complexity, additionally showing number and type of severe issues to be solved first.

For more details, please refer to Section *MES Test Manager® (MTest)*.

17.3 CUSTOM PAGES

A third way to look at the data in a detailed way is to create an arbitrary number of additional pages containing user-defined trend and status charts for selected measures, i.e. base measures, derived measures, as well as quality properties.

The way how to structure and to visualize the imported data to be analyzed in that way, is described in the *Custom Pages* chapter.

18 STATUS REPORT

MQC allows the creation of a status report documenting and summarizing the most important data as visualizations (taken from the MQC pages) and tables (focusing either on Artifacts or Quality properties), so that the information is displayed in detailed form.

The report is designed for browsing (HTML) and print (PDF).

To create a new Report, click on the *Create* button in the *Create Report* dialog in the menu on the left-hand side.

Note: It is important to know that the active settings from the filter panel are applied for the creation of the report, opening up a more indepth customization of the displayed data.

Active markings, in contrast to filters, are not applied in the creation of the report.

18.1 CONFIGURATION OF THE STATUS REPORT

The report configuration dialog allows an extensive configuration of the contents contained in the report. On one side the artifacts contained in the report can be selected by different criteria. On the other side the amount of information for these artifacts may be limited.

18.1.1 Artifact selection

If you want to limit the artifacts contained in the report, MQC provides several use cases that can be configured in the following way:

- **All: Include all artifacts in report** All artifacts are contained inside the status report. These artifacts are sorted from worst to best by default.

The following options allow users to limit those artifacts for which the status report will be generated:

- **By Quality Bin: Bad** Show only artifacts with severe quality problems, i.e. all your artifacts with a very low quality (marked in MQC in red) are added to your report.
- **By Quality Bin: Acceptable and Bad** Show only artifacts with “yellow” and “red” quality (exclude good quality results), thus concentrating on results that must be improved considerably.

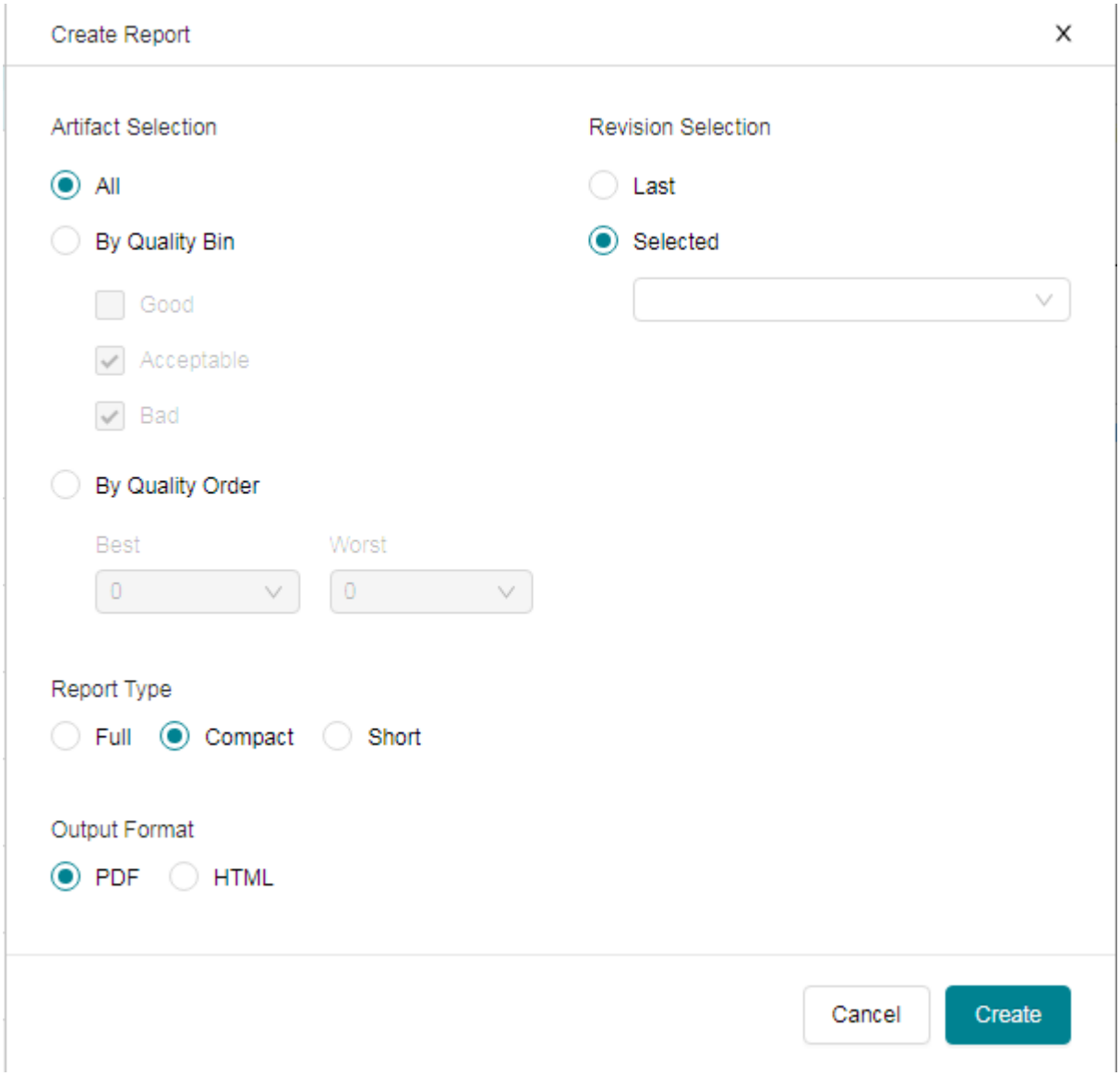


Figure 18.1: The Create Report dialog allows the configuration of specific settings for the report.

- **By Quality Bin: Good and Acceptable** Show only artifacts with "green" and "yellow" quality (exclude bad quality results), thus concentrating on artifacts with at least acceptable quality.
- **By Quality Bin: Acceptable** Show only artifacts with a "yellow" quality, i.e. all your artifacts with an aggregated quality percentage value which ranges from 20 to 80.
- **By Quality Bin: Good** Show only artifacts with a "green" quality, i.e. all your artifacts that are already quite good.
- **By Quality Order: Best "x" and worst "y" (artifacts)** Show the "x" artifacts with the best and "y" artifacts with the worst quality at the same time. You can compile your artifacts in such a way that the desired amount of artifacts with the best and/or worst quality can be exported within your report.
- **By Quality Order: Worst 5 (artifacts)** Show only a given number (e.g. 5) of worst artifacts (by quality). This option allows you to create a report with a stable number of worst (or best) artifacts regardless of its classification according to the aforementioned Quality Bins (Good, Acceptable, and Bad).

Furthermore, in case you have to take control of a huge amount of artifacts we recommend combining the functionalities within the Status Report generation with the wide range of possibilities provided by the filter panel. For example, given the case that you have defined an Artifact Structure, you will be able to create a report with specific (Good, Worst, Best, Acceptable, etc.) artifacts of only one (or multiple) StructureGroups or StructureElements *Artifact Structure*).

18.1.2 Revision selection

A status report is for exactly one revision. The last revision is selected by default, but any other revision can be chosen in this dialog. Please, note that trend charts will appear in the status report for the whole project duration (or the filtered duration/revisions according to the filter panel).

18.1.3 Report Type

Three different report types are available to choose from: Full, Compact, and Short. Every report type displays relevant data and visualizations of the following categories.

Table 18.1: Included content in the report for the different report types

	Full	Compact	Short
Quality Overview	✓	✓	✓
Data Availability Overview	✓	✓ (*)	✓ (*)
Data Trend Overview	✓	-	-
Data Status Overview	✓	-	-
Artifact Details	✓	✓	-
Quality Property Details	✓	✓	-
User Configured Custom Pages	✓	✓	✓
Configuration Information	✓	-	-

* Does not contain the “Missing Base Measures per Artifact” table.

18.1.4 Output Format

You can chose between two different output formats namely PDF or HTML.

After clicking on *Create* the report will open within the default viewer on your platform, where you can save it from.

18.2 CONTENT OF THE REPORT

18.2.1 Report and Project Information

The Report Information shows the configurations of the report and the creation date. In addition it contains the project name, information regarding the due date of your milestones, the overall availability value and overall quality value.

The Project Information is a summary of the project for the selected revision.

18.2.2 Quality Overview

This section contains the most important visualizations and information of the quality pages in MQC, such as the Quality Distribution, Quality Status for the selected revision, Quality Trend by Artifact and the Quality Trend by Property.

Additionally, quality status information for the selected revision per Artifact as well as per Quality Property are available in tables.

18.2.3 Data Availability Overview

In the Data Availability Overview the Availability Distribution shows the data availability over time.

The Availability Matrix and the Availability Status is shown only for the selected revision of the report.

Special attention is paid to missing data. Not yet imported base measures are listed per artifact and data source. Please note, this information is included only when the report type “Full” is selected.

18.2.4 Data Trend Overview

For the Data Trend Overview the measure trend by artifact visualizations presents the base measure values per artifact across all revisions.

18.2.5 Data Status Overview

The Data Status Overview provides a data status visualization for each data source for the selected revision.

18.2.6 Artifact Details

The Artifact Details section shows, per artifact, the Quality Trend as well as the Quality Status for all quality properties. The Quality Status is only shown for the selected revision.

Additionally a table for the selected revision with the quality property values is included, which can also be used to directly navigate to the correlating part of the Quality Property Details.

18.2.7 Quality Property Details

The Quality Property Details section shows the Quality Trend by Artifact as well as the measurement function, which is used to calculate the quality from the imported data for each quality property.

Additionally a table is shown that displays the data for all measures inside the measurement function for each artifact, as well as the file source from which the data was read. The table cells inside this table are either displayed white for available, blue for propagated or grey for missing.

Much like in the Artifact Details, this table can be used to directly navigate to the corresponding part of the Artifact Details section.

18.2.8 User Configured Custom Pages

The user configured custom pages are dependent on the configured pages in MQC (for more details refer to *Custom Pages*). The status and trend visualizations for each page are shown.

18.2.9 Configuration Information

This section provides an overview for all imported configurations divided into Quality Configurations, Structure Configurations and Other Configurations.

The Quality Configurations include the Quality Model Configuration, the Derived Measure Configuration and the Base Measure Configuration.

The Structure Configurations include the Artifact Mapping Configuration, the Project Milestone Configuration and the Context Category Configuration.

The Other Configuration includes the user configured Custom Page Configuration.

19 ANALYSIS SETUP CONFIGURATION

MQC provides the possibility of loading an setup configuration to easily setup an analysis. The setup configuration is done via an Excel file. Within this file the user is able to define

- files and directories containing reports of tools/data sources to be imported into MQC
- names and paths of MQC configuration files, e.g. project structure
- as well as almost all options, e.g. revision granularity.

19.1 HOW TO USE

After unziping sample data you can find an example for an setup configuration in it with the name of **00_SetupConfiguration.xlsx**, configured for MQC showcase.

Type	Value
BasePath	
ProjectStructure	01_ProjectStructures_Initial.xlsx
QualityModel	02_QualityModel_Initial.xlsx
TargetValues	05_TargetValuesPerMilestone.xlsx
CustomPages	06_ReportPagesConfig_Indiv_TestPage.xlsx
Data	Data
Data	
Data	
Data	
Data	
AutoRefresh	false
Propagation	ALL:Project
RevisionGranualarity	CalendarWeek
ShowEmptyRevisions	false
EnableContextCategories	true
ExcludeEmptyValues	false
ShowTargetValues	true
PagesQuality	Quality Status,Quality Sunburst,Quality HeatMap,Quality Trend
PagesData	Data Availability,Data Status,Data Trend
PagesTool	MXAM, MTest, MXRAY, Polyspace
PagesCustom	Target Page,AK Measure Report

Figure 19.1: Sample of setup configuration file

For using this feature, first copy sample data anywhere in your local file system or network then open MQC and instead of choosing **Tools/New Analysis**, import **00_SetupConfiguration.xlsx** by clicking on `Browse local file...`.

Note: If you are using web version of MQC, before importing setup configuration you need to ensure

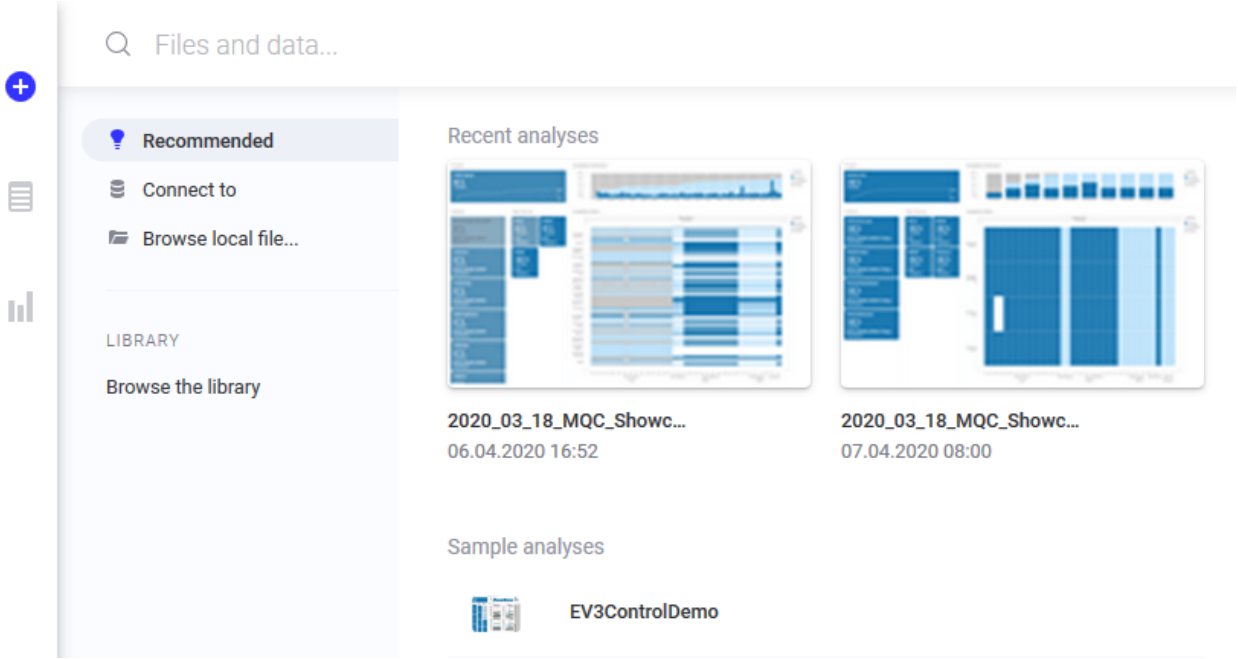


Figure 19.2: How to import Setup Configuration excel file for easy setup MQC

that the MQC showcase is located on a network drive which is mounted at and can be accessed by the server. Open the **00_SetupConfiguration.xlsx** and update the **BasePath** value with the network path of the directory where the MQC showcase is located.

19.2 CONFIGURATION

The setup configuration is an excel file with two columns named `Type` and `Value`. Contents of first column (Type) are constant and second column should be configured.

This configuration can be divided in three parts:

1. Set the paths of all files that can be imported in MQC. This contains:

- Project structure configuration (see *Project Structures Configuration*)
- Quality model configuration (see *Quality Model Configuration*)
- Target values configuration (see *Target Values*)
- Custom pages configuration (see *Custom Pages*)
- Files and directories containing reports of tools/data sources: for each of directories and single files to be imported, one row is added to the setup configuration file via type `Data` and path of them as value.

Note: As Figure 19.1 shows, the first row of the setup configuration is for setting a **BasePath**. If all files are located under the same path, this can be defined one time as **BasePath** and all other path values may be

configured as relative paths.

2. Set default values for MQC options. This contains:

- Automatic refresh: can be true or false (see *Configuration*)
- Propagation: can be None, All:Project or All:Milestone (see *Data Propagation*)
- Revision granualarity: can be Month, CalendarWeek or Day (see *Revision granularity*)
- Show empty revisions: can be true or false (see *Project Milestone Structure*)
- Enable context categories: can be true or false (see *Add whitelists/ blacklists to your Context Categories*)
- Show target values: can be true or false (see *Target Values*)

3. Set the pages, which should be shown in the analysis. For each row from a group of provided pages in MQC, chosen pages should be added and separated by **comma (,)**. For more detaile information please see *Pages*

Note: All values may be left empty. MQC applies defaults for all values, which are is not configured here.

Errors in configuration will be ignored. In any case an analysis is created and the user gets a notification via a validation dialog after the load of the intial configuration has been finished.

Contact

Mail:

Model Engineering Solutions GmbH

Waldenserstraße 2-4

10551 Berlin

Germany

Email: info@model-engineers.com

Tel.: +49 (0) 30 2091 6463 0

Fax: +49 (0) 30 2091 6463 33

Web: https://model-engineers.com/

TECHNICAL support

Web: https://model-engineers.com/mqc.html

Email: mqc@model-engineers.com

Tel.: +49 (0) 30 2091 6463 50

Copyright Note

This document contains proprietary information that is protected by copyright. All rights are reserved. Neither the documentation nor the software may be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written consent of Model Engineering Solutions GmbH.

© 2014-2020: Model Engineering Solutions GmbH, Waldenserstraße 2-4, 10551 Berlin, Germany

THIRD-PARTY PRODUCTS

MATLAB®, Simulink®, Stateflow® and Embedded Coder® are registered trademarks of The MathWorks, Inc. TargetLink® is a registered trademark of dSPACE GmbH. Windows®, Excel® and Visual Studio® are registered trademarks of the Microsoft Corporation. Testwell CTC++ is a trademark of Verifysoft Technology GmbH.