Software Quality Objectives for Source Code
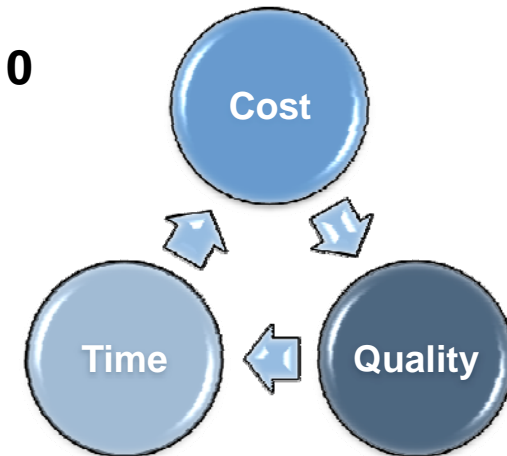
A project with Renault, PSA, Valeo, Delphi, MathWorks

MathWorks Automotive Conference – June 23rd 2010

<u>Presenters:</u>

Thierry Cambois - PSA PEUGEOT CITROËN

Patrick Munier - MathWorks

Agenda

1. Objective and History of the project
2. Results
3. Current Status
4. Perspectives

Agenda

1. Objective and History of the project
2. Results
3. Current Status
4. Perspectives
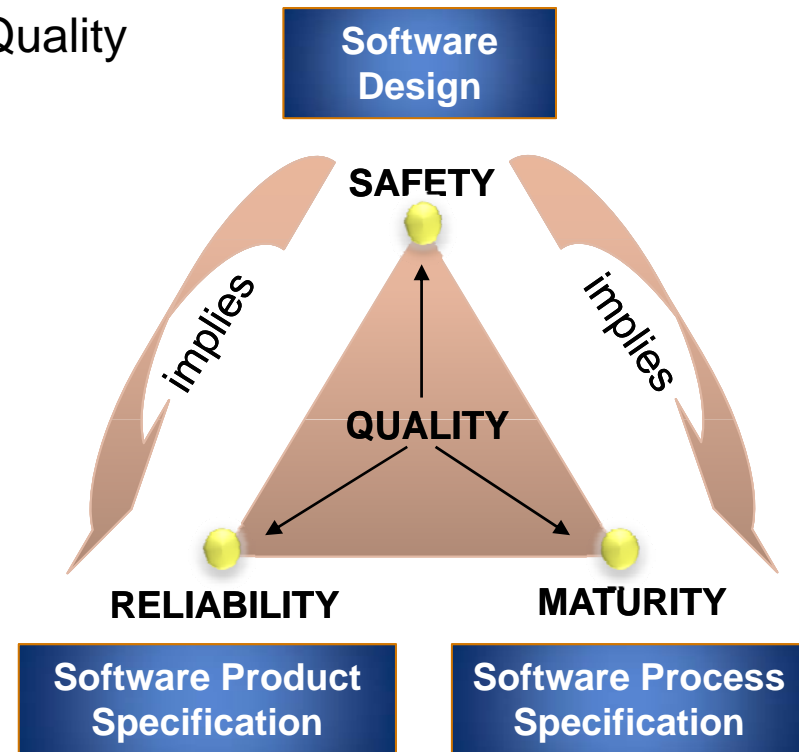
Automotive Manufacturers and Software Quality

- ## Examples of Quality Assurance Objectives

  1. **Prevent** Quality Issues of Embedded Software Products

  2. **Control Deliveries** (Product and Process)

  3. **Audit/Evaluate** Embedded Software Quality

  4. **Investigate** Root Causes of Defects

- ## Policy

  - Establish Standards about Embedded Software **Product** and **Process**

  - Use semantically correct software products which **prove** that the software is **reliable**

  - Develop according to **mature** software processes

Software Design

SAFETY

implies

QUALITY

implies

RELIABILITY

MATURITY

Software Product Specification

Software Process Specification

Project Overall Objective

- From a discussion on how to **optimize the use of PolySpace**, the following question was raised:

  How to **formalize relationships** between the French Automotive manufacturers (Renault, PSA) and their suppliers, with respect to a set of **software quality objectives**?

  - ➤ Need to produce a common **document template** for the communication between Automotive manufacturers and their suppliers, <u>focused on the goal to achieve</u>

  - ➤ Need to produce **guidelines on using PolySpace** with regard to this template, <u>focused on the means to establish</u>
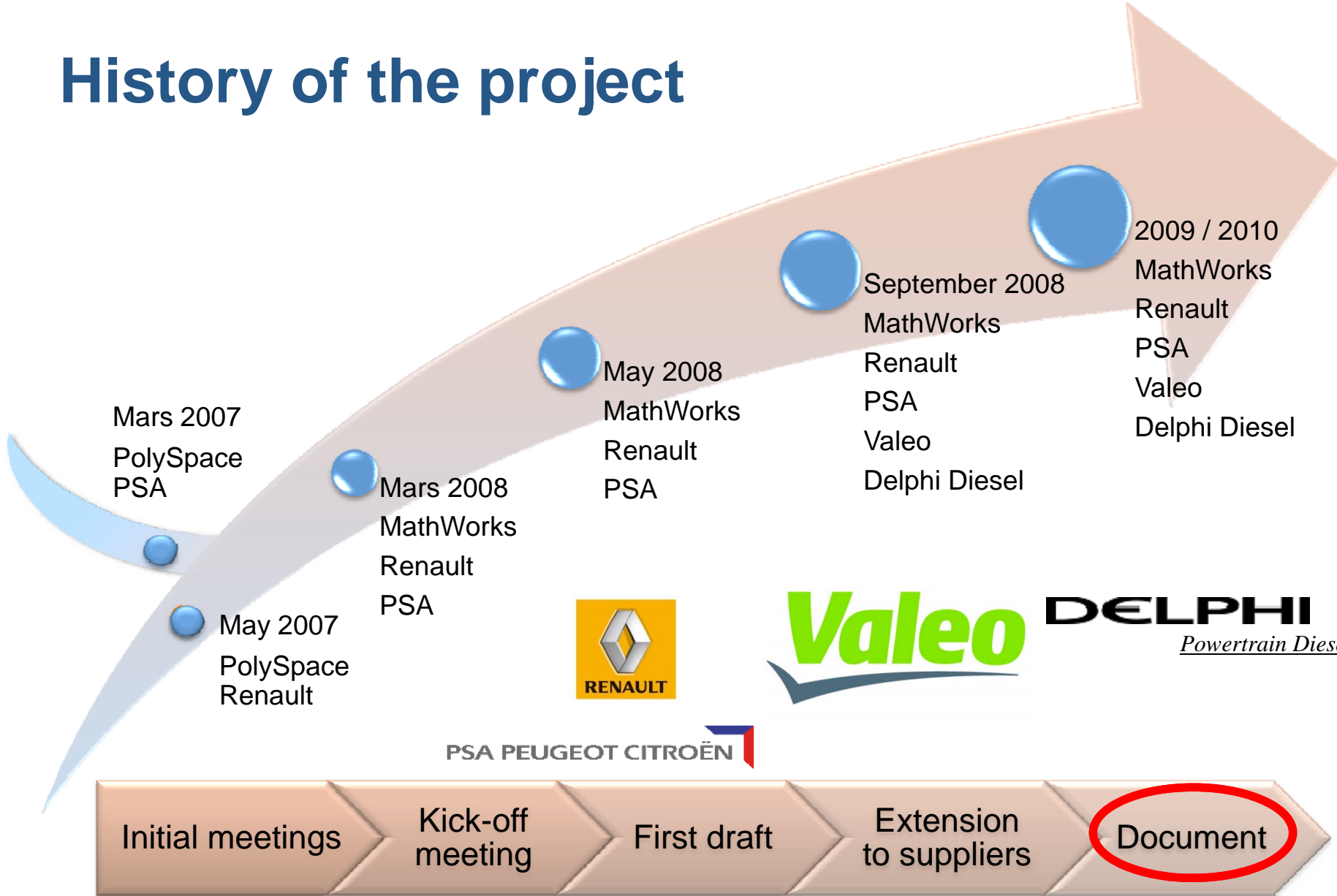
  ➡ **Clarification, Buy-in, Agreement**

History of the project

Mars 2007
PolySpace
PSA

May 2007
PolySpace
Renault

Mars 2008
MathWorks
Renault
PSA

May 2008
MathWorks
Renault
PSA

September 2008
MathWorks
Renault
PSA
Valeo
Delphi Diesel

2009 / 2010
MathWorks
Renault
PSA
Valeo
Delphi Diesel

Valeo

DELPHI
Powertrain Diesel

RENAULT

PSA PEUGEOT CITROËN

| Initial meetings | Kick-off meeting | First draft | Extension to suppliers | Document |

RENAULT PSA PEUGEOT CITROËN The MathWorks™ **Valeo** **DELPHI**

Agenda

1. Objective and History of the project
2. **Results**
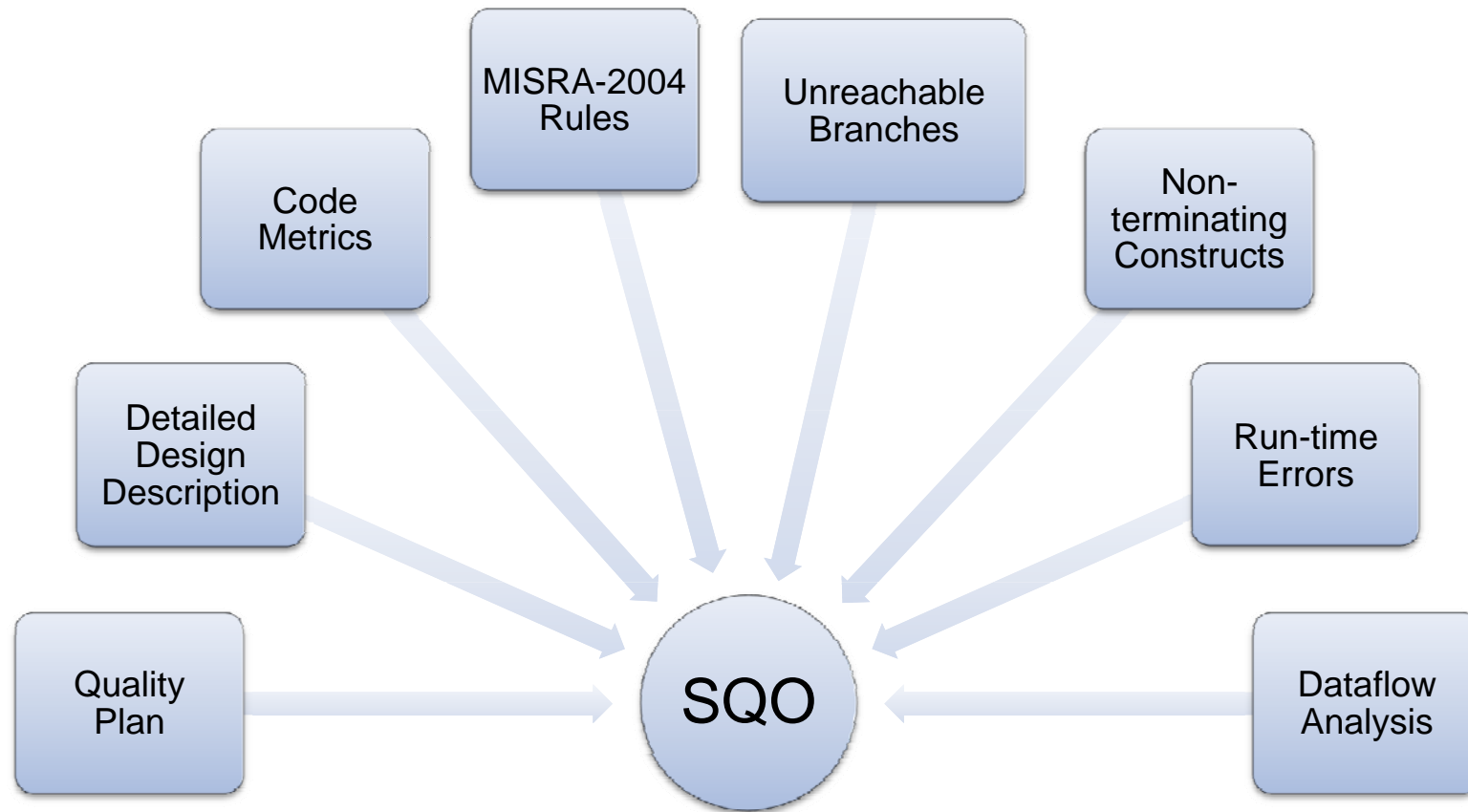3. Current Status
4. Perspectives

Software Quality Objectives : Key Categories

Software Quality Objectives : Incremental Quality

SQO Step 1
- Quality Plan & Detailed Design
- Code Metrics
- 1st MISRA-2004 rules subset

SQO Step 2
- Systematic run-time errors
- Non terminating constructs

SQO Step 3
- Unreachable branches

SQO Step 4
- 1st subset of potential run-time errors

SQO Step 5
- 2nd MISRA-2004 rules subset
- 2nd subset of potential run-time errors

SQO Step 6
- 3rd subset of potential run-time errors
- Dataflow Analysis

SQO Level 1
SQO Level 2
SQO Level 3
SQO Level 4

Steps define Process & Product Requirements for Modules
**Levels define expected Steps Progress for Deliveries**

Software Quality Objectives : Deployment Process

- A Supplier delivers an application with 3 modules
  ➔ What is the process?

  1. Define the different deliveries for the application
  2. For each module of the application, specify its target Quality Level
  3. Assign Software Quality Steps to intermediate deliveries
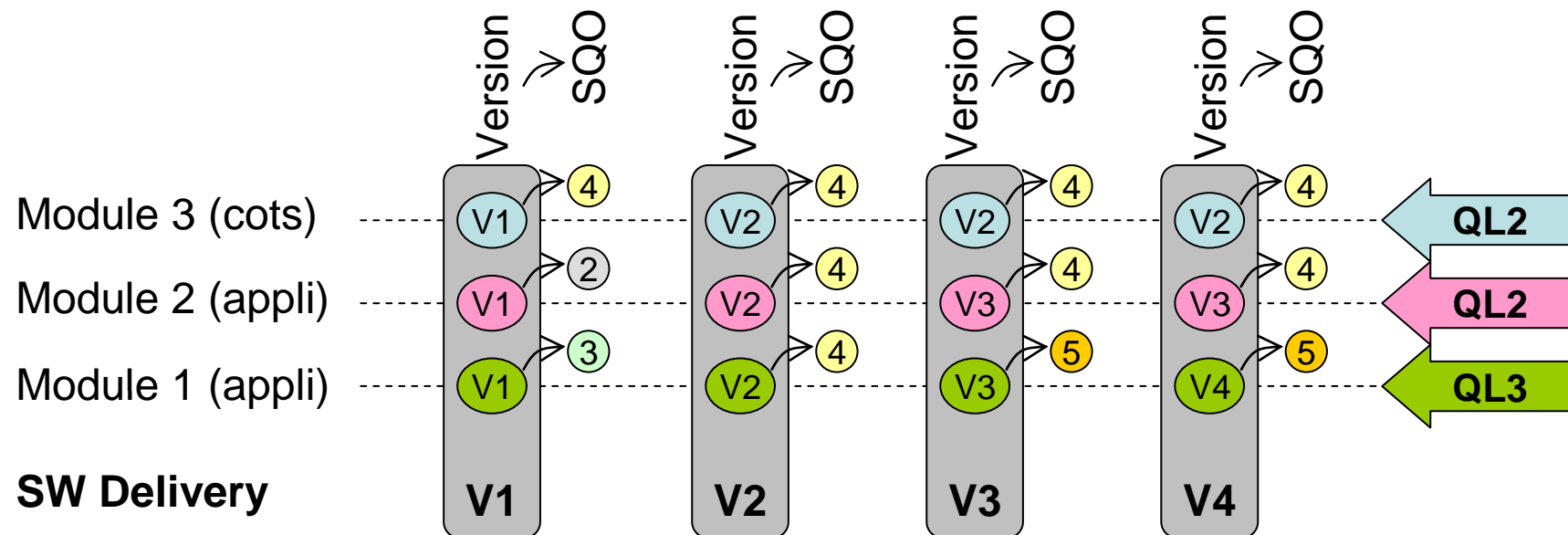
27 Software Quality Requirements

- **SQR-50:** The supplier shall provide the list of tools and methods used

- **SQR-70:** The supplier shall justify that methods and tools used are appropriate to achieve the requirements

- **SQR-140:** The automotive manufacturer and the supplier shall choose at the beginning of the project the code <u>code metrics that will be used</u>

- **SQR-150:** For the chosen metrics, the supplier shall demonstrate that the modules <u>comply with the agreed boundary limits</u>, or <u>justify the deviations</u>

- **SQR-160:** The supplier shall demonstrate that all the files within a module are compliant with the "first MISRA rules subset". The supplier shall correct or justify all violations of the rules

- **SQR-200:** The supplier shall demonstrate that for all files within a module <u>a review of systematic runtime errors has been performed</u> and that <u>errors which have not been corrected are justified</u>, for the following categories: out-of-bound array access, …
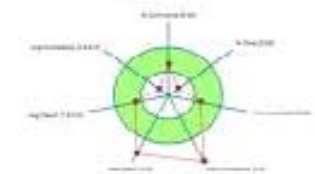
Agenda

1. Objective and History of the project

2. Results

3. Current Status

4. Perspectives

Current Status

- Version 2.0 of the document is available

- PSA, Renault (France) and Nissan (Japan) integrated the SQO document in their Software Quality Requirements

- Hyundai (Korea) is considering using SQO

- Delphi Diesel has integrated these new requirements in their process

- Valeo begins to use the document internally

- Raised interest of other industry communities such as Railway Transportation

Agenda

1. Objective and History of the project
2. Results
3. Current Status
4. Perspectives
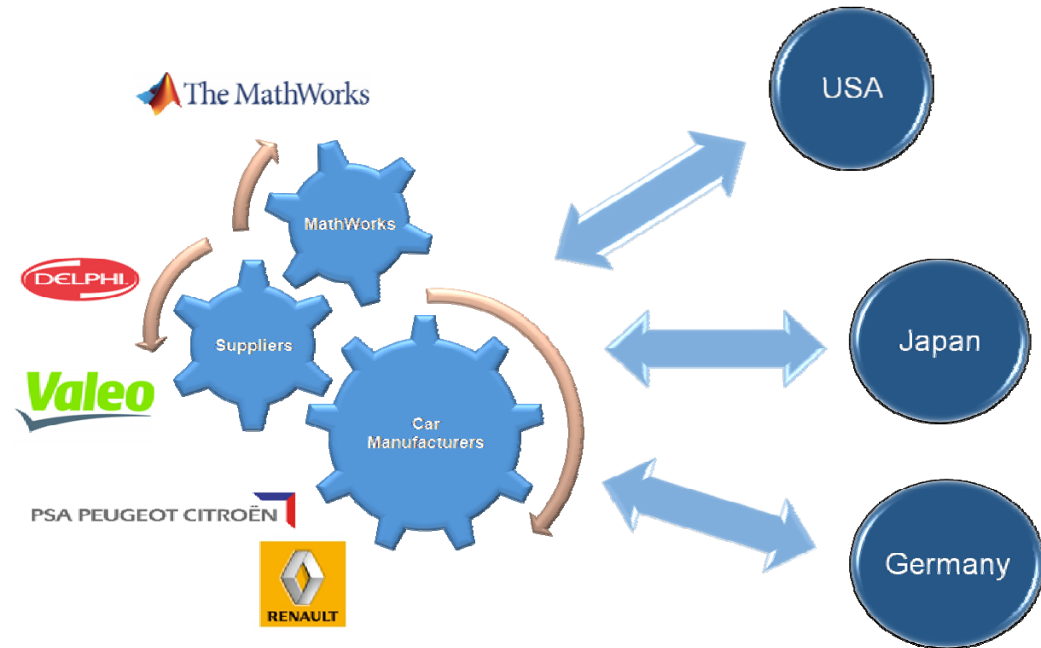
Perspectives

- Use of SQO by other car manufacturers

- Use of SQO by Automotive COTS Vendors

- Use of SQO by other industries

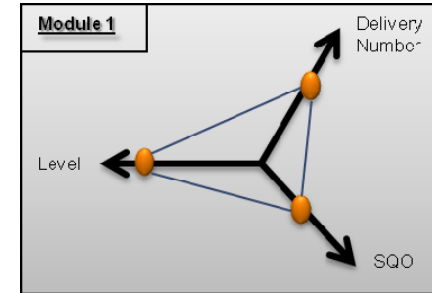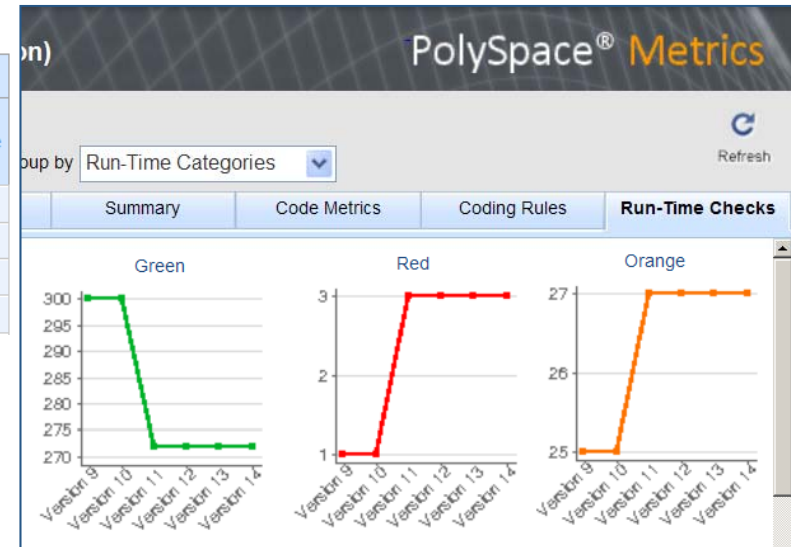 - Avionics

  - Railway Transportation

Conclusion - The SQO document is

- **Adaptive** to the context

- **Available** and being used

- **Win/win** for suppliers and manufacturers

- **Aligned with ISO-26262** Standard objectives

- A **guideline** for future versions of PolySpace

Verification	Coding Rules		Run-Time Errors		Software Quality Objectives					
	Confirmed Defects	Violations	Confirmed Defects	Run-Time Reliability	Overall Status	Level	Review Progress	Code Metrics over Threshold	Justified Coding Rules	Justified Run-Time Errors
2.0 (2)				100.0%	PASS	SQO-3	0	0	0	0
2.0 (1)				98.1%	FAIL	SQO-3	0	1	0	0
1.8			2	92.3%	FAIL	SQO-3	1	1	0	0
1.7	1	2	3	85.1%	FAIL	SQO-3	3	1	0	0

Guideline on using PolySpace products with regards to SQO:
http://www.mathworks.com/matlabcentral/fileexchange/27525

Thanks for your attention

Any question?