Experiences with SQIL – SW Quality Improvement Leadership Approach from Volkswagen

Richard Messnarz¹, Maik Sehr², Ingrid Wüstemann², Joachim Humpohl³, Damjan Ekert¹

¹ISCN GesmbH, rmess@iscn.com, dekert@iscn.com
²BOSCH Automotive Steering maik.sehr@bosch.com
³HELLA Joachim.Humpohl@hella.com

Abstract. SQIL stands for SW Quality Improvement Leader. This is a new schema by Volkswagen where improvement agents are trained at suppliers to trace the quality of the processes, the quality of the product, and a set of metrics to track the progress of the development. The idea is to create a synergetic collaboration between suppliers and Volkswagen. Volkswagen defined so called KGAS (Konzerngrundanforderungen Software, in English: Group Basic Requirements Software) requirements which extend the scope of Automotive SPICE by a number of consistency requirements for engineering. The SQIL knows these requirements, usually has an Automotive SPICE Assessor qualification and correctly interprets the VW additional criteria to coach the improvement programs.

Keywords: Improvement, KGAS, SQIL, Automotive SPICE ®

1 The Need for an Extension of Automotive SPICE by Additional KGAS criteria

Vehicles nowadays are not only understood as a set of components to be integrated but as a set of vehicle functions which are linked with sub-functions/features at component level [3]. A steering system, for instance, delivers the correct movement of the steering rack and angle position of wheels and sends the steering angle on the bus, the ESP uses this input (besides more inputs like speed, yaw rate, etc.) to slow down the inner wheels, the second axle steering system uses the steering angle to set a supporting steering angle at the second axle, the torque vectoring uses that input to speed up the outer wheels, the active damper system configures the right characteristic curve based on that steering angle, speed etc. to keep the car stable in the curve, and so forth. So it is usually not possible to assign one function to only one component. Vehicle functions define a sequence of component functions/features and provide a bus specification with defined messages and timing to support the real time sequence of interactions of component functions/features. For manufacturers it is important to manage this complexity, at VW this principle is called FUN (function-oriented development).
In the next development step of ADAS (Autonomous Driving Assistant), test tracks are currently established, e.g., a test track at A2 in Graz, Austria [9] cloud functions will be added to this schema so that the functional hierarchy will even get one level more complex. When driving a car on a specific road section, for instance, the cloud will deliver the typical steering angle, which other cars used, the steering angle of neighbouring cars, information if obstacles were identified by cameras of cars who went through the road section in the last 10 minutes, etc. The cloud can also set a defined speed and steering angle in full self-driving mode.

**Safety Goal:** Do not steer more than requested by the command. Commands then include a requested steering angle, this is translated in the ECU to a requested torque and the achieved angle position (internal steering angle) is then compared with the external requested steering angle [2],[4],[9].
**Why Functional Safety is not enough:** If a system is developed by ASIL D (based on ISO 26262) the electronic and software has to prove an architectural metric, in form of a 10 FIT (Failure in Time rate based on operating hours for single point faults). 1 Fit is $10^{-9}$ and 10 FIT means 1 hazardous fault in $10^8$ operating hours in the fleet.

Usually a passenger car is driven in the lifetime $10^4$ hours. So far ca. 35 million Golf cars were produced, makes $35 \times 10^6$. Let us assume that only the last 2 generations of Golf cars are counted and we use approx. 10 million cars, i.e. $10 \times 10^6 = 10^7$. The fleet has then $10^7 \times 10^4 = 10^{11}$ hours. That means that even when developing the whole design based on ASIL – D requirements $10^3$ (1000) hazardous faults would appear in the field with that fleet size. Therefore VW needs a rigorous design approach which even extends the scope of Automotive SPICE and functional safety [2],[5],[6],[7],[8],[9].

**The Quality of the Requirements Drives the Quality of the System:** In KGAS (version 3.0, from 3.11.2015, [3]) there are a number of requirements about how to describe a requirement, extending the scope of Automotive SPICE. The underlying idea is that requirements are verifiable (clear input condition and result that can be tested), atomic (e.g. clear functional decomposition), unambiguous (e.g. no different interpretations possible), measurable, etc. Also certain attributes for the analysis are required by KGAS, such as risk (e.g. assigned ASIL level), feasibility (feasibility is described in the new base practice “Evaluate” in Automotive SPICE 3.0, with guidance in the VDA yellow book for the interpretation of ASPIE 3.0).

**KGAS examples:**
- KGAS_3193: All requirements must be evaluated in terms of risks and feasibility.
- KGAS_3247: All requirements must be unambiguous.
- KGAS_3248: All requirements must be self-consistent.
- KGAS_3249: All requirements must be understandable.
- KGAS_3250: All requirements must be feasible.

Moreover, KGAS asks you to formulate requirements with a clear IF <condition> THEN <event> syntax that helps to fulfil the above mentioned criteria.

**KGAS example:** KGAS_3267: Each specification of system and software requirements and software as well as system and software elements and components must follow a defined schema (e.g. [Condition] [the system or software or component] [shall or must do] [action or procedure or interface requirement]).

**A Strategy assure that all engineers work towards the same process goals:** In VW assessments, it is important to present a how-to-guide or process plan document with a strategy per process. Meanwhile a VDA yellow book [10] (1st edition, February 2017) provides guidance about how to rate the strategy base practice in SWE.4 SW
Unit Verification to SWE.6 SW Qualification Test, SYS.4 System Integration and Integration Test to SYS.5 System Qualification Test, SUP.1 Quality Assurance, SUP.8 Configuration Management, SUP.9 Problem Resolution Management, and SUP.10 Change Request Management processes. Example Checklist for SW Unit Test Strategy SWE.4 BP1:

The expectations for a software unit verification strategy cover these aspects:

a) A definition of the verification objects
b) A definition how specific requirements regarding verification and testing (e.g. test-specific stakeholder requirements, ISO 26262, HIS-Metrics, MISRA) are covered.
c) A definition of the methods for test case and test data development derived from the detailed design and the non-functional requirements (e.g. development of positive/negative tests, equivalence class analysis).
d) A definition of the methods and tools for static verification and for reviews.
e) A definition of the test environment regarding each test method
f) A definition of the test coverage in relation to project and release plan
g) A definition of entry and exit criteria for the software unit verification
h) A documentation of sufficient test coverage of each test level, if the test levels (e.g. software qualification test, software integration test and unit test) are combined
i) An approach for the handling of failed tests, failed static verifications (e.g. justification for failed MISRA-check or compiler warnings) and review findings.

Fig. 4. VDA Yellow Book – Checklist for strategy underlying SWE.4 BP1

Moreover, the Yellow book [10] defines dependencies so that down rating BP1 automatically leads to further down rating of other BPs (Base Practices).
Breaking down to a manageable size of a unit and traceability In the KGAS [3] the idea is that, the linking concept (see Fig. 1) allows to trace SW requirements to a manageable size of a code block in the source code. Usually this would be e.g. a C-function or a block (with input – output transitions) in a model based development with defined maximum size of the source code.

KGAS examples: KGAS_3322: For software units with cyclomatic complexity smaller or equal than 10 bidirectional traceability between the unit (at least header level) and the software detailed design must be available. KGAS_3323: For software units with cyclomatic complexity greater than 10 bidirectional traceability between the software unit elements and the software detailed design must be available.

The cyclomatic complexity tells how many paths are possible through the software. Every if, else, else if, switch-case statement increases the cyclomatic complexity. This figure therefore tells how many test vectors are needed as a minimum to cover all paths. The above definition means that a cyclomatic complexity 10 is manageable and the unit can be linked as a whole, when the unit is more complex the inside of the unit must be linked upward (with blocks inside the unit that again should be lower than 10 cyclomatic complexity).

There are many more such system engineering principles which are the root cause for KGAS requirements. In total, the KGAS Version 3.0 from November 2015 contains more than 400 such additional requirements.

2 The SQIL Report

The report of the software quality as part of the SQIL [3] activities should include:

a) The fulfilment of the process quality according to Automotive SPICE®
b) Findings of the requirements consistency check
c) Metrics indicating the progress of the product development and test coverage.

The SQIL report should reported regularly (monthly) to the quality department of Volkswagen. In the next chapters, each of the activity will be explained in more detail.

2.1 Process reviews and KGAS

Most of companies have already implemented some kind process reviews according to process checklists, where they check if the project activities are compliant with their process. In Automotive SPICE® 3.0 this is reflected with following base practice: SUP.1.BP3: Assure quality of process activities. Perform the activities according to the quality assurance strategy and the project schedule to ensure that the processes meet their defined goals and document the results.
When performing process reviews, the additional KGAS requirements, can be incorporated in the existing checklists, lowering the effort of performing additional reviews against the KGAS. In the below example, the question referring to the project schedule is extended to fulfil the KGAS requirements:

**Process review question:** *The project schedule is updated weekly and consistent with the project milestones*

**Relevant KGAS requirements:**

- KGAS_3164: A schedule based on the project structure plan must be created
- KGAS_3176: The Schedule must be based on the effort according to the estimations of activities
- KGAS_3173: Each activity planned in the schedule must have a start date, end date, duration, and effort, degree of fulfilment, resources and dependencies
- KGAS_3182: The schedule must contain the software, hardware, customer, functional safety and quality assurance milestones.
- KGAS_3184: The schedule must not contain any activities with a duration longer than a working week
- KGAS_3597: The schedule must not contain any activities with an effort higher than a working week

**Combined process review question:** *The project schedule is updated weekly and consistent with the project, system and subsystem milestones. All activities are planned with an effort and duration of less than 40h and contain the degree of fulfilment, an assigned resource and dependencies.*

The process reviews should be performed according to the KGAS_3477 at least each 2 months and reported to the quality department of Volkswagen.

### 2.2 Consistency check of requirements

With Automotive SPICE® 3.0 [1] the traceability and consistency of requirements are addressed by separate base practices. Where the traceability refers to the links between requirements, design, code, and test cases, the consistency addresses content and semantics [9]. In Fig. 6, the bidirectional traceability and consistency concept is shown:
Traceability is usually measured and tracked by using metrics and trends, where the consistency of requirements can be checked only with reviews. In projects even with a high test coverage (100% traceability from requirements to tests), problems and bugs are found due to poor written requirements or test cases only covering parts of the requirements. Therefore one of the goals of the SQIL review is also to check the content of the requirements, design, code and test cases and if they are consistent to each other. A typical review starts by selecting a customer function/sub-function in the customer requirements specification. The links are then followed to the requirements specification, architecture, and detail design documents to the code section, continuing with links to the test cases. The consistency from the code to the requirements is checked by selecting a function and following the links through to the initial requirement. Each step should be documented with a screenshot, the path to the work product and the identifier of the objects or work product (e.g. requirement SYS_ID182, detailed design for the function …). The findings should be then confirmed by an expert.

Fig. 6. Bidirectional Traceability and Consistency [10]
2.3 Monitoring by trends of performance

Many of the metrics collected and reported to VW base on the traceability model in Annex D of Automotive SPICE 3.0. The annex D [1] describes the key concept criteria for understanding Automotive SPICE ® (see Fig. 6).

When working in VW projects a dashboard must be created, filled with data, and reported as a trend. Most of the data base on the traceability aspects in Automotive SPICE (see Fig. 4). The measurements are done monthly and once you have two measurement points you can calculate a vector, with three measurement points a trend can be estimated. The coverage metrics are rated with symbols such as:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>⬇️</td>
<td>Performance is decreasing</td>
</tr>
<tr>
<td>⬆️</td>
<td>Performance is increasing</td>
</tr>
<tr>
<td>⬇️→</td>
<td>Coverage is the same as before</td>
</tr>
</tbody>
</table>

The SQIL collects the following data in reviews and evaluate the performance:

Number based metrics:
- No. of System/SW Features planned
- No. of System/SW Features implemented
- No. of System/SW Features positively verified
- No. of Customer requirements implemented
- No. of Customer requirements implemented and linked to system requirements
- No. of Software related System requirements reviewed
- No. of SW related System requirements linked to system element(s)
- No. of SW related System requirements linked to SW requirements
- No. of SW requirements reviewed
- No. of SW requirements linked to SW elements
- No. of SW requirements linked to SW units
- No. of SW components linked to SW detailed design
- No. of SW component interfaces specified in SW architecture
- No. of SW units in SW specified in detailed design
- No. of SW units implemented according to SW detailed design
- No. of SW units verified
- No. of SW units positively verified
- No. of SW units in SW detailed design linked to SW unit test cases
- No. of SW component interfaces verified
- No. of SW component interfaces positively verified
- No. of SW component interfaces specified in SW architecture linked to SW integration test cases
- No. of SW requirements verified
- No. of SW requirements positively verified
- No. of SW requirements linked to SW test cases
- No. of System interfaces verified
- No. of System interfaces positively verified
- No. of System interfaces linked to System integration test cases
- No. of System requirements verified
- No. of System requirements positively verified
- No. of System requirements linked to System test cases
- Number of all customer requirements

Coverage based metrics for development process:
- Impl. Coverage: Customer Requirements
- Review Coverage: System Reqs.
- Linking Coverage: Sys Reqs. --> SW Reqs.
- Review Coverage: SW Reqs.
- Linking Coverage: SW Reqs. --> SW Arch.
- Linking Coverage: SW Reqs. --> Units
- Linking Coverage: SW components --> SW DD
- SW units impl. Specified in DD
- Impl. Coverage: SW units acc. DD spec

Coverage based metrics for testing process:
- SW Units verified
- SW Units positively verified
• SW units in DD linked to test cases
• SW component interfaces verified
• SW component interfaces positively verified
• SW component interfaces linked to SW integration test cases
• SW Requirements verified
• SW Requirements positively verified
• SW Requirements linked to SW Test Cases
• System interfaces verified
• System interfaces positively verified
• System interfaces linked to sys. Integration test cases
• System Requirements verified
• System Requirements positively verified
• System Requirements linked to System Test Cases

In Volkswagen projects requirements are grouped to features (most suppliers called it function; e.g. lane assist function of a steering system) and the performance per function/feature is tracked. Fig. 7 shows an actual template for a functional performance evaluation used at BOSCH AS.

![Fig. 7. Performance Evaluation per Function/Feature](image)

### 3 Experiences and Hints from Industry Partners

#### 3.1 Bosch AS Experience with KGAS Implementation

**Automating the Metrics is Important:** BOSCH AS (Automotive Steering Systems) developed in the VW MQB project an automated Q-Report in 2010, which was then used in VW projects until 2013 and then rolled out to all projects in the company. The Q-report already contained the traceability and coverage metrics for requirements and
test levels and has two variants. In the “released” report, only released requirements are counting and in the “development” report, all analysed requirements are counting.

From 2013 to 2016 the report has been further refined. The Volkswagen data are then generated from the data of the Q-report plus additional data from the development environment. Experience shows that such data must be automated so that 2-weekly or latest monthly a progress overview can be generated.

**Achieving a High Level of Coverage by a Baukasten / Tool Kit:** In the VW MQB project a Baukasten of requirements and design elements as been set up with a set of re-usable links. System functions have been agreed among all steering system variants and a re-usable mother of a system specification has been set up. System requirements are grouped by system functions. SW functions are defined in a Baukasten SW concept and SW functions relate to SW components and SW units. SW requirements have been grouped by SW functions. Also standard system test catalogues have been created for vehicle verification, test bench and HIL test and re-usable test specifications are linked with the requirements. Similar it works on SW level where for each SW function there is a directory with SW requirements, design requirements, HIL test specifications etc. and completely linked. If you select a function you drag all requirements, test specifications, etc. into the project (a child derived from a mother module).

This Baukasten is adaptable to different projects by a placeholder {...} concept where in the text of the requirement a measurable parameter (e.g. {max-voltage}) is named generically within brackets and in a data file the project enters the value (e.g. 30 V) and then the requirements specification generates the right text with the project specific value. This way the same specification in a generic way can be adapted by parameters. Also due to the Baukasten concept ca. 86% of links have been re-usable.

**Tracking Safety and Non-Safety Coverage Separately:** The development of steering systems with a servo motor is highly safety critical and it includes safety goals with an ASIL-D classification (highest possible safety integrity level in Automotive). The automated Q-report of BOSCH AS generates three types of trend metrics, one for the overall coverage, one for safety and one for non-safety coverage. This means that, for instance, for the system test coverage the Q-report displays the coverage of all system requirements as a percentage, the coverage of all safety classified (ASIL-A to ASIL-D) requirements in system test, and the coverage of non-safety relevant (e.g. with QM classification) system requirements.

This strategy is used to create a synergy between VW technical audits, functional safety assessments, and ASPICE assessments.

**Using the Dashboard Strategy in all Projects as a Standard:** The VW Dashboard was first introduced at Bosch AS in 2014. A large number of the VW Dashboard metrics have been taken directly from the Q-Report (developed in the VW MQB project in 2009 to 2013). It was possible to re-use a large number of metrics from the Q-report by small adaptions / changes of the Q-Report to provide the needed metrics to VW.

Currently BOSCH central quality is establishing a standard for a dashboard (BBM metrics) which will be used for all projects concern wide and agrees this dashboard with Volkswagen. In BOSCH AS first external SQILs were used in 2015, and since
2016 in BOSCH AS internal SQILs have been trained to support the process and dashboard optimization, and more SQILs will be trained in 2017.

3.2 HELLA Experience with KGAS Implementation

Using the Dashboard and Trends to Track the Quality Performance: In HELLA the standard VW dashboard has been implemented and standard data and trends are reported monthly. Quality managers with an Automotive SPICE assessor background are trained as a SQIL and act as improvement coordinator in VW projects.

Some of the dashboard trends are explained in the below Figs. 8 and 9. However, these figures are only an example and do not display a real project. Fig. 8 displays a feature coverage trend. In VW projects, the software and system requirements are assigned to features. Features are planned by integration releases and the maturity of features is tracked.

Fig. 8 displays a trend of software and system requirements implementation coverage. In VW projects, this implementation trend is to be tracked to display the progress in the development.

![Feature Tracking](image)

**Fig. 8.** Tracking the maturity of features/functions in total

![System & Software Requirements Implementation](image)
Using a Process Compliance Coverage Trend

In HELLA a standard process review tool is used to rate the compliance of each process in the project at least quarterly. Based on that tool a process coverage trend can be tracked. In addition, the process review checklist is based on ASPICE and additional VW specific KGAS questions.

4 Outlook

Volkswagen is currently promoting a new version of the dashboard concept, which defines the documentation required by SQILs. This includes:

1. Process Check
2. Consistency Check
3. Monitoring Metrics

Fig.10 shows a process check where the field WP Quality is linked with specific metrics in the dashboard. Project management, for instance is related to the customer requirements implementation and linking. The progress of plan is linked with reaching the agreed target coverage, and the trend field shows if the related metrics show an increase, decrease or no change in the performance.

The concept of consistency check was described in chapter 2.2 of the paper. In addition, the monitoring metrics have been listed in chapter 2.3 of the paper.
5 Relationship with the SPI Manifesto

The SPI manifesto [11] has been developed in 2009 at an international workshop attached to EuroSPI 2009 in Alcala, Spain. It describes 3 values and 10 principles which make SPI really work.

The following principles of the SPI manifesto have a direct relationship with the SQIL approach.

5.1 Base Improvements on Experience and Measurement

Section 2.3 describes a number of metrics used to track the progress of a development project in the Automotive area (based on Automotive SPICE 3.0 and VW KGAS requirements).

5.2 Support the organisations needs and vision

Section 1 of the paper describes a number of system engineering paradigm which are specific for working in Volkswagen projects. They reflect the understanding of the organisation when working in the organisation.

Some of them are (see section 1):

- The FUN principle.
- The FUN principle will be extended with self driving cars.
- The Quality of the Requirements Understanding with clear KGAS criteria which criteria must be met.
- Etc.

In general the KGAS [3] requirements represent a VW (adapted for projects in this organisation) specific understanding of the needs when using Automotive SPICE.

5.3 Apply Risk Management

Section 1 of the paper describes the paradigm “Why Functional Safety is not enough” together with a risk calculation why the additional VW KGAS requirements are needed.

Also the monitoring approach and dashboard are used to display a progress trend and based on the trend a projected performance is calculated. The biggest risk (with a he financial impact) in Automotive projects is the delay of an SOP (Start of Production). This trend based metric approach allows projects to anticipate risks by projected trends and the illustration of expected performance.

5.4 Ensure all parties agree and understand the process

Section 2.1 of the paper describes the process reviews approach where quarterly reviews are done for all processes in scope (based on Automotive SPICE) and a
process performance trend is created during the project lifetime showing the improvement in the processes until full coverage is reached.

References