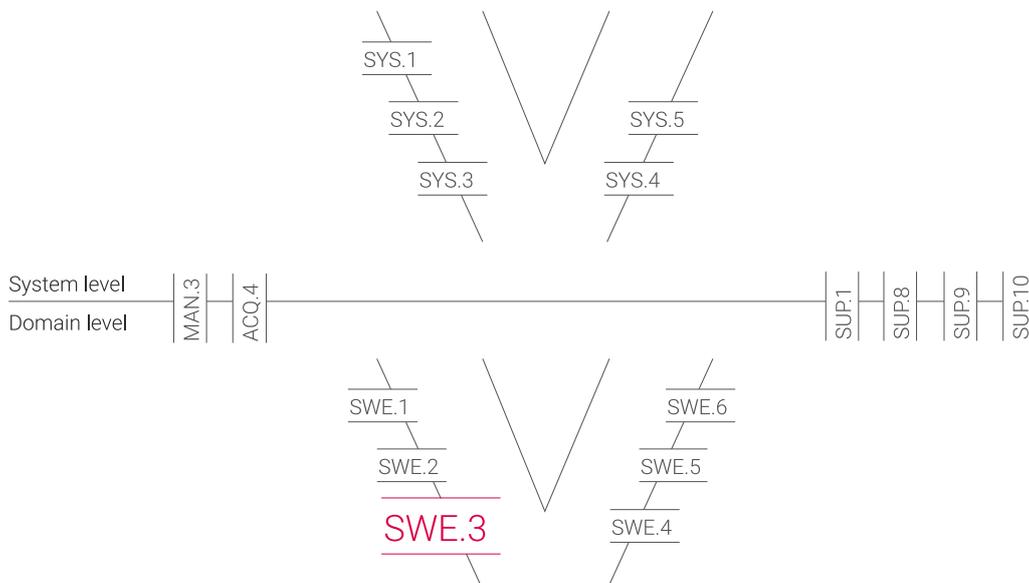


SOFTWARE DETAILED DESIGN AND UNIT CONSTRUCTION – SWE.3

in Automotive SPICE®

an introduction

Dr. Bhaskar Vanamali



Learn Automotive SPICE® with Kugler Maag Cie GmbH

1st version – April 2021



Competence from A to Z: Training courses and workshops

Whether you need training to gain a certificate or an individual workshop, we can set up and provide the training you need, anything from courses for experts to exclusively in-house training at your place of business. Workshops or training courses? With us, you'll find it a lot easier!



Who to contact?

Smilja Mateja
Training Coordination
+49 7154 1796 229

information@kuglermaag.com

About this white paper

This white paper extends the materials I've covered in my Automotive SPICE® tutorial for Beginners on YouTube.

<https://www.kuglermaag.com/swe3>

<https://youtu.be/HhhHEtZWj4k>

Both the YouTube tutorial and this document cover the core concepts and they are not complete by any means. This publication has been prepared for general guidance only. Please do not act according to any information given in this document without receiving specific professional consultancy. The publisher, KUGLER MAAG CIE GmbH, shall not be liable for any damages resulting from any use of the information contained in this report.

If you want to learn everything Automotive SPICE® has to offer and become an Automotive SPICE® expert, check out our ASPICE Training:

<https://www.kuglermaag.com/training/automotive-spice.html>



Competence from A to Z: Training courses and workshops

Whether you need training to gain a certificate or an individual workshop, we can set up and provide the training you need, anything from courses for experts to exclusively in-house training at your place of business. Workshops or training courses? With us, you'll find it a lot easier!



Who to contact?

Smilja Mateja
Training Coordination
+49 7154 1796 229

information@kuglermaag.com

About the author



Bhaskar Vanamali is Principal and Partner at Kugler Maag Cie GmbH. He has been working on process improvement for nearly 20 years and was secretary of the working group 13 of VDA QMC.

He is Principal Assessor and Trainer for Automotive SPICE[®], and a co-author of books. He has performed more than 140 assessments and trained more than 250 assessors.

Due to his background he is trying to shed light on new approaches from different perspectives. He is actually a veterinary by profession but is working in IT since 24 years.



Competence from A to Z: Training courses and workshops

Whether you need training to gain a certificate or an individual workshop, we can set up and provide the training you need, anything from courses for experts to exclusively in-house training at your place of business. Workshops or training courses? With us, you'll find it a lot easier!



Who to contact?

Smilja Mateja
Training Coordination
+49 7154 1796 229

information@kuglermaag.com

Short summary of the YouTube tutorial

The Software Detailed Design and Unit Construction process in Automotive SPICE® (also known as SWE.3) helps your organization to provide an evaluated detailed design for the software components and to specify and to produce the software units.

What is the goal of the Software detailed design? A lot of organizations and projects have problems understanding how to document the detailed design. So, what are the three aspects you should manage in this process?

Aspect One: level of detail

Often, organizations struggle how specific the detailed design should be. A good way to approach this is to remember what the purpose of the detailed design is. It is the basis for the implementation of the code and the unit test. Especially the unit verification requires a detailed description. Here, the level of coverage must be considered.

In safety-critical software ISO 26262 provides guidance. In non-safety software typically at least C0- or statement coverage, by some customers C1- or branch coverage is required. The higher the coverage goal the more detail is required by the detailed design. If you have an ASIL-B classified module a C1-coverage is expected. That means that your detailed design should identify the different branches of your software.



Competence from A to Z: Training courses and workshops

Whether you need training to gain a certificate or an individual workshop, we can set up and provide the training you need, anything from courses for experts to exclusively in-house training at your place of business. Workshops or training courses? With us, you'll find it a lot easier!



Who to contact?

Smilja Mateja
Training Coordination
+49 7154 1796 229

information@kuglermaag.com

Aspect two: Interfaces

A pitfall I often encounter in assessments is the lack of detailed description of the external and internal interfaces.

Expected content of interface documentation are

1. Names
2. Types
3. Units
4. Resolutions
5. ranges and
6. default-values.

Without this information a proper testing of the interfaces in the unit test is impossible.

It is okay if the external interfaces are described on architectural level and tested in the software integration test.



Competence from A to Z: Training courses and workshops

Whether you need training to gain a certificate or an individual workshop, we can set up and provide the training you need, anything from courses for experts to exclusively in-house training at your place of business. Workshops or training courses? With us, you'll find it a lot easier!



Who to contact?

Smilja Mateja
Training Coordination
+49 7154 1796 229

information@kuglermaag.com

Aspect three: when shall the detailed design be documented

Describe the detailed design before you implement the code.

Often, the detailed design is described after the fact, meaning after the code has been written. Why is that a problem? The unit test should check whether the code fulfils the detailed design.

If you write the detailed design after documenting your code, the point of the unit test is lost. Now, you could argue that you may not need the unit test. The point is that you should derive your detailed design from your architecture and not from the code. If the chain is broken, then suddenly the content of documentation and all the tests do not make sense anymore.

So, the detailed design must be written before you start writing your code. There is nothing wrong to iteratively develop detailed design and code step by step.



Competence from A to Z: Training courses and workshops

Whether you need training to gain a certificate or an individual workshop, we can set up and provide the training you need, anything from courses for experts to exclusively in-house training at your place of business. Workshops or training courses? With us, you'll find it a lot easier!



Who to contact?

Smilja Mateja
Training Coordination
+49 7154 1796 229

information@kuglermaag.com

Software Detailed Design and Unit Construction – the process according to Automotive SPICE®

The purpose of the Software Detailed Design and Unit Construction Process is to provide an evaluated detailed design for the software components and to specify and to produce the software units.

BP1: Develop software detailed design. Develop a detailed design for each software component defined in the software architectural design that specifies all software units with respect to functional and non-functional software requirements.

BP2: Define interfaces of software units. Identify, specify and document the interfaces of each software unit.

BP3: Describe dynamic behavior. Evaluate and document the dynamic behavior of and the interaction between relevant software units.

NOTE 1: Not all software units have dynamic behavior to be described.

BP4: Evaluate software detailed design. Evaluate the software detailed design in terms of interoperability, interaction, criticality, technical complexity, risks and testability.

NOTE 2: The results of the evaluation can be used as input for software unit verification.

BP5: Establish bidirectional traceability. Establish bidirectional traceability between software requirements and software units. Establish bidirectional traceability between the software architectural design and the software detailed design.



Competence from A to Z: Training courses and workshops

Whether you need training to gain a certificate or an individual workshop, we can set up and provide the training you need, anything from courses for experts to exclusively in-house training at your place of business. Workshops or training courses? With us, you'll find it a lot easier!



Who to contact?

Smilja Mateja
Training Coordination
+49 7154 1796 229

information@kuglermaag.com

Establish bidirectional traceability between the software detailed design and software units.

NOTE 3: Redundancy should be avoided by establishing a combination of these approaches that covers the project and the organizational needs.

NOTE 4: Bidirectional traceability supports coverage, consistency and impact analysis.

BP6: Ensure consistency. Ensure consistency between software requirements and software units. Ensure consistency between the software architectural design, the software detailed design and software units.

NOTE 5: Consistency is supported by bidirectional traceability and can be demonstrated by review records.

BP7: Communicate agreed software detailed design. Communicate the agreed software detailed design and updates to the software detailed design to all relevant parties.

BP8: Develop software units. Develop and document the executable representations of each software unit according to the software detailed design.

Output Work Products: Software Detailed Design, Software Unit, Communication Record, Review Record, Traceability Record



Competence from A to Z: Training courses and workshops

Whether you need training to gain a certificate or an individual workshop, we can set up and provide the training you need, anything from courses for experts to exclusively in-house training at your place of business. Workshops or training courses? With us, you'll find it a lot easier!



Who to contact?

Smilja Mateja
Training Coordination
+49 7154 1796 229

information@kuglermaag.com

Advanced tutorial about Software Detailed Design and Unit Construction

What is the benefit of Software Detailed Design and Unit Construction?

The software detailed design is the link between the software requirements, software architecture, and the implementation. This is done in iterative steps:

- The software architecture describes how the software is organized into components and how they interact.
- The software detailed design fully details each component and describes the structure and behavior of the software units.
- The software units are created according to the detailed design specifications.

The detailed design provides a complete specification for the software developer to construct the required software unit. This specification helps to ensure that only high quality and traceable tested software units are being developed.



Competence from A to Z: Training courses and workshops

Whether you need training to gain a certificate or an individual workshop, we can set up and provide the training you need, anything from courses for experts to exclusively in-house training at your place of business. Workshops or training courses? With us, you'll find it a lot easier!



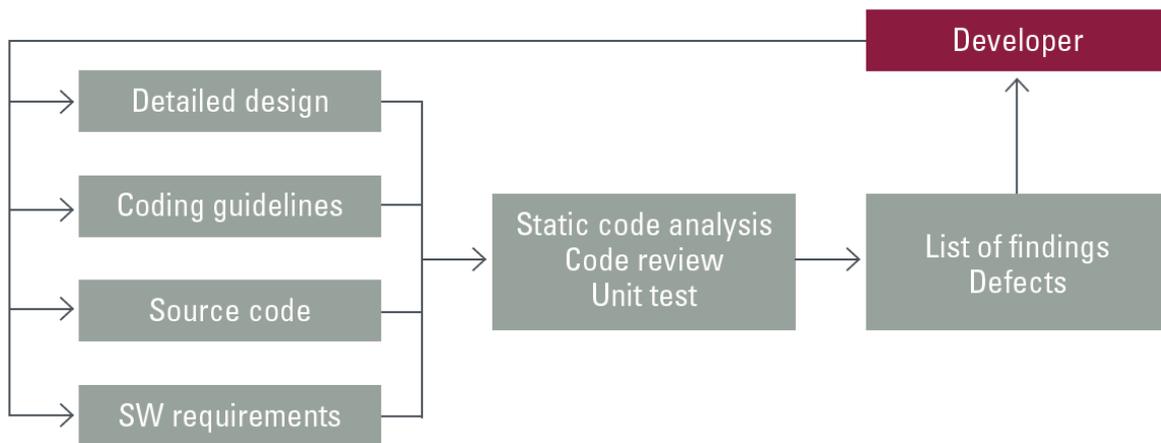
Who to contact?

Smilja Mateja
Training Coordination
+49 7154 1796 229

information@kuglermaag.com

What is the content of the Software Detailed Design and Unit Construction process?

- Based on the software architecture the detailed design for each software component is created and documented. It describes the required logic and algorithms, internal and external interfaces, dynamic behaviors, and error conditions (BP1, BP2, BP3)
- The detailed design is evaluated and verified, ensuring it supports the software architecture (BP4)
- The traceability between the software requirements and software units, software architecture and the software detailed design, and the detailed design and software units are ensured and checked for consistency (BP5, BP6)



Methods of unit verification

- The detailed design and any changes to it are agreed upon and communicated to the relevant parties (BP7)



Competence from A to Z: Training courses and workshops

Whether you need training to gain a certificate or an individual workshop, we can set up and provide the training you need, anything from courses for experts to exclusively in-house training at your place of business. Workshops or training courses? With us, you'll find it a lot easier!



Who to contact?

Smilja Mateja
Training Coordination
+49 7154 1796 229

information@kuglermaag.com

- The source code of a software unit is created according to the detailed design specifications. In model-based development it will be generated from the model (BP8)

Experiences, problems and hints:

- How the software architecture and the detailed design are structured: The architecture consists of »elements« which can be broken down into lower-level elements. The elements on the lowest level are called »components«. Detailed design starts with these components and breaks them down into »units« which have atomic character.
- What a unit is, is often not clearly defined or consistently used. For example, is a unit a C file, a method or function, or an atomic element in a model?
- A detailed design specification for software component usually includes the algorithms used, defined interactions and interface specifications with input and output variables as well as description of the program structure. Often we see flow and state diagrams, message sequence charts, UML diagrams, and natural language descriptions used to help describe portions of the detailed design.
- Source code comments are an essential part of the unit.
- Development must be carried out taking into account the coding guidelines and standards. Compliance is ensured by static code analysis and code reviews.
- Historically, detailed design is often over-simplified or skipped entirely. Teams have tried to document the design through the use of markup



Competence from A to Z: Training courses and workshops

Whether you need training to gain a certificate or an individual workshop, we can set up and provide the training you need, anything from courses for experts to exclusively in-house training at your place of business. Workshops or training courses? With us, you'll find it a lot easier!



Who to contact?

Smilja Mateja
Training Coordination
+49 7154 1796 229

information@kuglermaag.com

languages in the source code. This may have the problem of not properly considering and reviewing the design ahead of implementation.

- The traceability between detailed design and software units is often ensured through naming conventions.
- Redundancy in the traceability is not required. However, if the links run from requirements to architecture to detailed design to the single unit, care must be taken that no information gets lost in between.



Competence from A to Z: Training courses and workshops

Whether you need training to gain a certificate or an individual workshop, we can set up and provide the training you need, anything from courses for experts to exclusively in-house training at your place of business. Workshops or training courses? With us, you'll find it a lot easier!



Who to contact?

Smilja Mateja
Training Coordination
+49 7154 1796 229

information@kuglermaag.com

Want to become an Automotive SPICE® expert?

If you're serious about learning Automotive SPICE® I highly encourage you to participate to one of our Automotive SPICE® Courses.

Visit the following page:

<https://www.kuglermaag.com/training/automotive-spice.html>



Competence from A to Z: Training courses and workshops

Whether you need training to gain a certificate or an individual workshop, we can set up and provide the training you need, anything from courses for experts to exclusively in-house training at your place of business. Workshops or training courses? With us, you'll find it a lot easier!



Who to contact?

Smilja Mateja
Training Coordination
+49 7154 1796 229

information@kuglermaag.com