Journal of Advances in Developmental Research (IJAIDR)

E-ISSN: 0976-4844 ● Website: www.ijaidr.com ● Email: editor@ijaidr.com

How to Succeed in ASPICE Assessments

Krishna Chaitanya Mamidala

Independent Researcher
Novi, Michigan, United States of America.
mamidala.kc@gmail.com

Abstract:
Automotive SPICE (ASPICE) assessments are crucial for organizations aspiring to demonstrate capability maturity in automotive software development. Success depends on strategic preparation, strong evidence management, process discipline, and effective engagement with assessors. This paper outlines practical strategies for succeeding in ASPICE assessments, supported by figures and tables illustrating key success factors, evidence structures, and common pitfalls.
This paper also provides an example about the preparation of an assessment for the Automotive SPICE process area Software Architectural design (SWE.2) to achieve Capability Level 1. We go into the process, the expected deliveries and the view of assessors. Always keeping the idea in mind: What to do to get through an assessment successfully?

Keywords: ASPICE, process assessment, automotive software, capability levels, quality assurance, software engineering, process improvement.

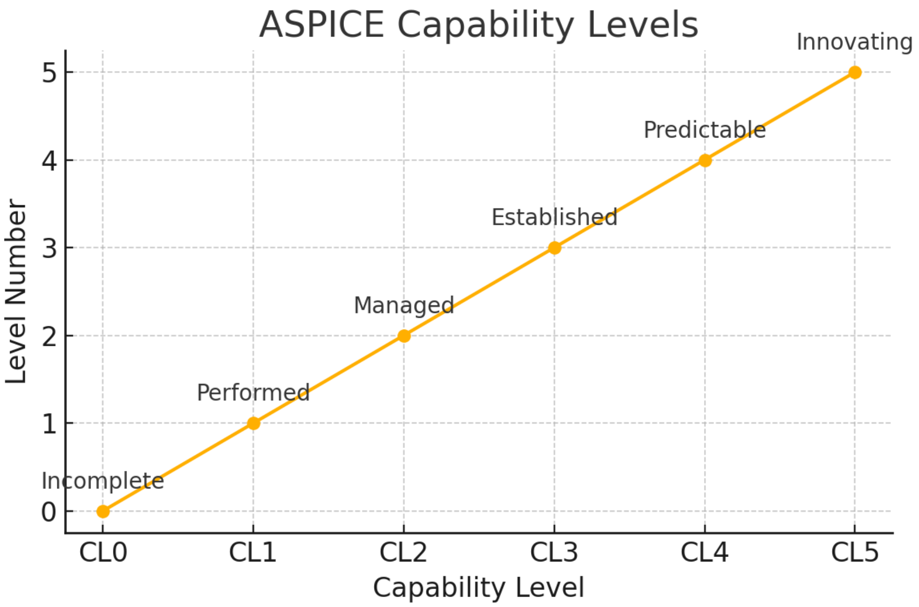
I. Introduction

The Automotive SPICE (ASPICE) framework is used globally to evaluate and improve software and systems development processes within the automotive industry. Achieving higher ASPICE levels directly influences an organization's market reputation, project opportunities, and risk management capabilities. This paper presents structured guidance for organizations aiming to excel in ASPICE assessments.
An Automotive SPICE Assessment is intended to determine the maturity level of an organization. The maturity level is seen as an indicator for high quality. The assessment itself is performed for each process using generic base practice descriptions derived from a reference process model. For a rating of Level 1: "performed process", at least 50% (largely) of all required achievements must be achieved.

2. Overview of ASPICE Assessment Model

ASPICE assessments evaluate both the process performance and capability using a set of predefined process areas and capability levels.

Figure 1. ASPICE Capability Levels and Key Focus Areas



3. Strategies for ASPICE Assessment Success

3.1. Build a Process-Oriented Culture

- Establish clear ownership of each ASPICE process.
- Integrate process activities naturally into daily project execution.

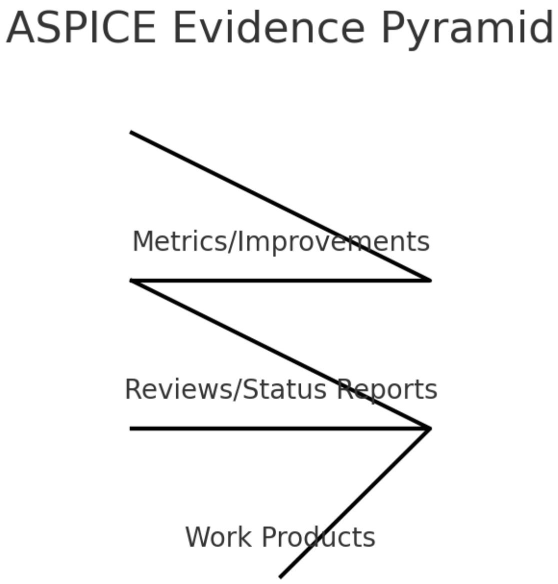
Table I. Sample Process Ownership Assignment

ASPICE Process	Process Owner	Key Work Product
SYS.2 System Requirements Analysis	System Engineer Lead	System Requirements Specification
SWE.2 Software Architectural Design	Software Engineer Lead	Software Architectural Design
SWE.4 Software Unit Verification	Software Test Engineer	Unit Test Reports
SUP.8 Configuration Management	Configuration Manager	Configuration Management Plan

3.2. Prepare Complete and Consistent Evidence

Focus on preparing the right evidence. Documents must be complete, consistent, and traceable.

Figure 2. ASPICE Evidence Pyramid



3.3. Conduct Internal Gap Analysis and Pre-Assessments

Perform early gap assessments. Benefits include:

- Identification of missing work products
- Staff readiness for interviews
- Early remediation of critical process gaps

3.4. Train Teams and Foster Awareness

Continuous education is critical. Conduct:

- ASPICE awareness sessions
- Role-based training on relevant base practices
- Workshops on interview handling during assessments

3.5. Master the Assessment Interview Process

Effective communication with assessors can heavily influence outcomes.

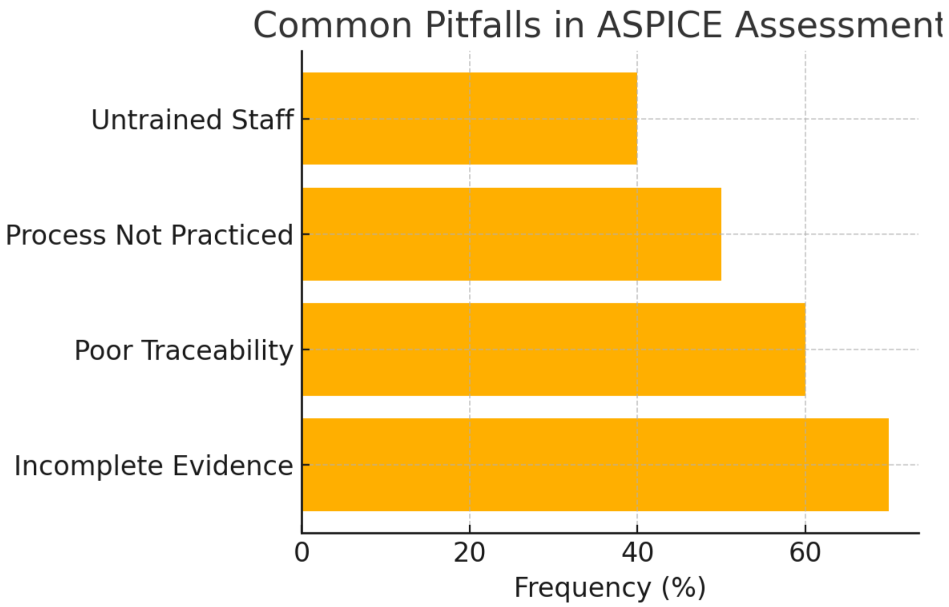
Table 2. Best Practices for Assessment Interviews

Do's	Don'ts
Provide clear, direct answers	Overload the assessor with unnecessary documents
Point assessors to specific evidence	Argue about the interpretation of standards
Be honest about gaps	Conceal known weaknesses
Highlight continuous improvement activities	Deflect or blame others for gaps

4. Common Pitfalls and How to Avoid Them

Many organizations face preventable pitfalls during ASPICE assessments:

Figure 3. Common Pitfalls in ASPICE Assessments



5. Preparation of Software Architectural Design (SWE.2) for an ASPICE assessment Capability Level 1

For every process area, Automotive Spice version 3.1 requires two basic types of deliverables:

- work products
- process outcomes

For SWE.2 process to be at Capability Level 1, the Process Outcomes and Work Products generated from this process needs to be evidenced during the assessment.

The purpose of the Software Architectural Design Process is to establish an architectural design and to identify which software requirements are to be allocated to which elements of the software, and to evaluate the software architectural design against defined criteria.

5.1 Process Outcomes

"As a result of successful implementation of this process:
1) a software architectural design is defined that identifies the elements of the software;
2) the software requirements are allocated to the elements of the software;
3) the interfaces of each software element are defined;
4) the dynamic behavior and resource consumption objectives of the software elements are defined;
5) consistency and bidirectional traceability are established between software requirements and software architectural design; and
6) the software architectural design is agreed and communicated to all affected parties.

5.2 Output Work Products

04-04 Software architectural design [OUTCOME 1, 2, 3, 4, 5]
13-04 Communication record [OUTCOME 6]
13-19 Review record [OUTCOME 5]
13-22 Traceability record [OUTCOME 5]
17-08 Interface requirement specification [OUTCOME 3]"

Table 3. Level1 Base practices for SWE.2 with requirements, Recommendations and Rules to comply with the Base Practices

Process Attribute	Base Practices	Activity, Outcomes requirements	Recommendations and Rules
PA 1.1	**Process performance process attribute**	The process performance attribute is a measure of the extent to which the process purpose is achieved. As a result of full achievement of this attribute: a) the process achieves its defined objectives.	
GP 1.1.1	**Achieve the process outcomes**	Achieve the intent of the base practices. Produce work products that evidence the process outcomes.	
BP1	**Develop software architectural design.**	Develop and document the software architectural design that specifies the elements of the software with respect to functional and non-functional software requirements. [OUTCOME 1]	[SWE.2.RL.1] If the software architecture does not reflect dynamic views the indicator BP1 shall be downrated. [SWE.2.RL.2] If the software architecture does not reflect applicable non-functional requirements the indicator BP1 shall be downrated.
BP2	**Allocate software requirements.**	Allocate the software requirements to the elements of the software architectural design. [OUTCOME 2]	Each requirement or requirement cluster is required to be mapped to at least one element of the software architectural design (“no requirement is forgotten”). - QMS Powertrain: non functional to sw elements, functional to sw components
BP3	**Define interfaces of software elements.**	Identify, develop and document the interfaces of each software element. [OUTCOME 3]	A software interface is defined by sender, receiver, format, size, resolution, quality information, frequency etc. of the data being transferred. On each layer of the static view of the software architectural design the interfaces between the elements are required to be identified.

			[SWE.2.RL.3] If the software interface definition is absent or incomplete regarding the definition above the indicator BP3 shall be downrated.
BP4	**Describe dynamic behavior.**	Evaluate and document the timing and dynamic interaction of software elements to meet the required dynamic behavior of the system. [OUTCOME 4]	Behavioral descriptions are required e.g. • state transition diagrams • sequence diagrams • message sequence charts • use-case diagrams [SWE.2.RL.4] If evidence of describing dynamic behavior regarding the topics mentioned above is missing the indicator BP4 shall be downrated. QMS Powertrain: Dynamic aspects to be described are: - the operating system including all possible modes and their transitions. - state machines of the software elements. - schedule, sequence and structure of tasks structure. - data flow between the software elements at least on those hierarchical levels which are relevant for the software integration.
BP5	**Define resource consumption objectives.**	Determine and document the resource consumption objectives for all relevant elements of the software architectural design on the appropriate hierarchical level. [OUTCOME 4]	[SWE.2.RL.5] If evidence of describing resource consumption objectives regarding the definition mentioned above is missing the indicator BP5 shall be downrated. QMS Powertrain The SW Architecture shall document the required response times and resources for memory (ROM, RAM, external / internal EEPROM or Data Flash) for the

			resource-critical elements of the software architecture.
BP6	**Evaluate alternative software architectures.**	Define evaluation criteria for architecture design. Evaluate alternative software architectures according to the defined criteria. Record the rationale for the chosen software architecture. [OUTCOME 1, 2, 3, 4, 5]	Variants, weaknesses, 1) Development of alternative solutions (e.g. for development of a completely new system) 2) Iterative architecture development 3) Carry over and adaption of an existing architecture (e.g. for platform development) [SWE.2.RL.6] If none of the three described approaches for architecture development is observable in the assessed project, PA1.1 shall be downrated. [SWE.2.RC.1] If the used procedure for architecture selection does not involve the required parties or competencies, the indicator BP6 should be downrated.
BP7	**Establish bidirectional traceability.**	Establish bidirectional traceability between software requirements and elements of the software architectural design. [OUTCOME 5]	[SWE.2.RL.7] If not all elements of the software architectural design are traceable to one or more requirement clusters, the indicator BP7 shall be downrated. The granularity is required to be respectively at least on the lowest granularity mentioned in the PAM: • single stakeholder requirement • single system requirement • single system architecture element • single software requirement • single software architecture component • single software detailed design element • single software unit • single verification criterion • single test case • single test result • single change request

			Recommendations and Rules: [T&C.RC.1] If the granularity is not at least on the lowest granularity mentioned above, the traceability indicator should be downrated.
BP8	**Ensure consistency.**	Ensure consistency between software requirements and the software architectural design. [OUTCOME 1, 2, 5, 6]	Review in process and review checklist
BP9	**Communicate agreed software architectural design.**	Communicate the agreed software architectural design and updates to software architectural design to all relevant parties. [OUTCOME 6]	On Capability Level 1 evidence for communication may consist of any tangible artifact (e.g. Emails, meeting minutes, voice recordings, etc.). The term “affected parties” is used here for the group of stakeholders who are directly processing the work products of a certain process in their work. Communication at Level 1 does not follow necessarily a plan or procedure. [Comm.RC.2] If there are evidences that necessary information is not provided to all relevant stakeholders (see examples in the list above), the indicator for “communicate agreed…” and/or the indicator for “summarize and communicate …” should be downrated.

6. Conclusion

Success in ASPICE assessments demands more than documentation—it requires real process maturity, strategic preparation, and clear communication. Organizations that embed quality in every stage of development, foster continuous improvement, and proactively manage assessments will consistently achieve higher capability levels and gain competitive advantages.

Automotive SPICE demands many activities and outcomes for quality assurance. Many of the required results should also be checked in a verifiable way. Knowing and applying these assessment rules increases the likelihood of reaching a good assessment. Usually, a project reaches level 1 after 2 years and level 2 after another 2 years. Experience shows that success is achieved most quickly when the team is willing to learn and works continuously to meet the requirements.

An Automotive SPICE Assessment is always successful if all participants from the project:

- have a good knowledge of the Automotive SPICE process Assessment model,
- can correctly answer questions from the assessor and
- can explain a relation of their activities to the reference process.

REFERENCES:

[1] Automotive Special Interest Group, Automotive SPICE® Process Assessment Model (PAM), Version 3.1, 2017.

[2] ISO/IEC 15504-2, Information Technology – Process Assessment – Part 2: Performing an Assessment, 2003.

[3] A. Radermacher, B. Wüst, "Success Factors for Automotive SPICE Assessments," Automotive Software Engineering Journal, 2019.

[4] https://www.synopsys.com/blogs/chip-design/preparing-for-automotive-spice-assessment.html

[5] ISO/IEC 33020:2015 Information technology -- Process assessment -- Process measurement framework for assessment of process capability

[6] ISO/IEC 15504-5:2006

Information Technology – Process assessment – Part 5: An exemplar Process Assessment Model

[7] Automotive SPICE® Guidelines, 2nd revised edition, November 2023