Assessing Agile in Automotive Embedded Development Projects Using Automotive SPICE 3.1

Richard Messnarz[1(✉)], Bernhard Sechser[2], Holger Höhn[2], Volker Lehmann[2], and Damjan Ekert[1]

[1] ISCN GesmbH, Graz, Austria
rmess@iscn.com
[2] Methodpark AG, Erlangen, Germany
bernhard.sechser@methodpark.de

Abstract. Automotive SPICE ® represents an assessment model which is used world-wide in mechatronic automotive projects where software is an essential part of the control system. In November 2018 a new version Automotive SPICE 3.1 has been published. This new version comes with an additional Automotive SPICE Guideline 1st Edition 2017 from VDA (available from Jan. 2018) which serves as a guideline to correctly interpret Automotive SPICE® 3.1. This Automotive SPICE Guideline for the interpretation of Automotive SPICE ® 3.1 includes rating rules and rating recommendations for agile projects. This article analyses these new rules and recommendations and illustrates how Automotive SPICE assessors will evaluate agile projects.

Keywords: Automotive SPICE 3.1 · Agile · Rating rules
Rating recommendations

1 Automotive SPICE 3.1 and Automotive SPICE Guideline Background

Figure 1 illustrates the VDA (Verband der deutschen Automobilindustrie - German Automotive Association) scope based on Automotive SPICE® 3.1 [1] which defines a set of processes for which an assessment is planned and performed.

For each process there is a set of base practices at capability level 1. Base practices contribute to process outcomes which help to achieve a process purpose (see MAN.3 example in Fig. 3).

In an assessment of capability level 1 (process attribute "process performance" in Fig. 2) the base practices (see Fig. 3) are rated with N (Not), P (partially), L (Largely) and F (Fully). The Automotive SPICE Guideline [2] provides rating recommendations and rating rules that are to be considered when performing the N/P/L/F ratings. The Automotive SPICE Guideline [2] also describes rating rules and rating recommendations for the higher capability levels 2 and 3.

X. Larrucea et al. (Eds.): EuroSPI 2018, CCIS 896, pp. 443–455, 2018.
https://doi.org/10.1007/978-3-319-97925-0_38

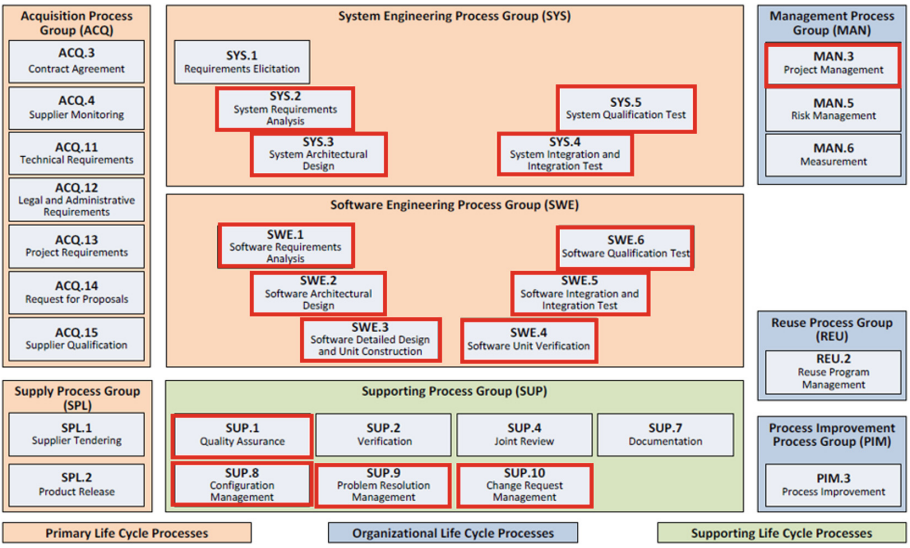

Fig. 1. Automotive SPICE 3.1 and VDA scope

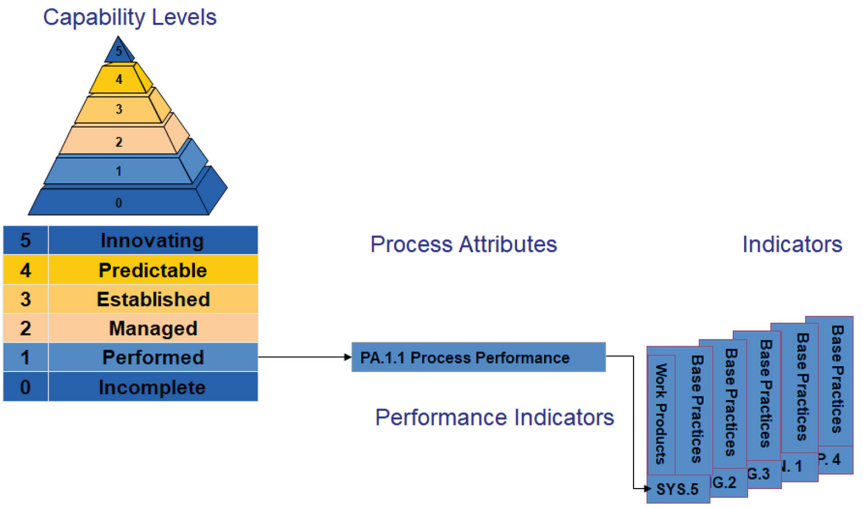

Fig. 2. Capability levels and base practices for level 1

The Automotive SPICE Guideline includes a chapter about agile developments [3, 4, 6, 17–21] and rating rules and rating recommendations in case of agile development [2].

Rating Scale: The ISO 33020 norm describes the rating scale N, P, L, F. Assessors rate at level 1 base practices and at higher levels (from level 2) generic practices and

assign evidences and comments to each indicator (base practice at level 1 and generic practice at level 2).

Figure 3 below shows all base practices of the MAN.3 process because later in the paper many of the agile relevant rules and recommendations reference some of these base practices.

Process ID	MAN.3
Process name	Project Management
Process purpose	The purpose of the Project Management Process is to identify, establish, and control the activities and resources necessary for a project to produce a product, in the context of the project's requirements and constraints.
Process outcomes	As a result of successful implementation of this process: 1) the scope of the work for the project is defined; 2) the feasibility of achieving the goals of the project with available resources and constraints is evaluated; 3) the activities and resources necessary to complete the work are sized and estimated; 4) interfaces within the project, and with other projects and organizational units, are identified and monitored; 5) plans for the execution of the project are developed, implemented and maintained; 6) progress of the project is monitored and reported; and 7) corrective action is taken when project goals are not achieved, and recurrence of problems identified in the project is prevented.

Base practices	
	MAN.3.BP1: Define the scope of work. Identify the project's goals, motivation and boundaries. [OUTCOME 1]
	MAN.3.BP2: Define project life cycle. Define the life cycle for the project, which is appropriate to the scope, context, magnitude and complexity of the project. [OUTCOME 2] *NOTE 1: This typically means that the project life cycle and the customer's development process are consistent with each other.*
	MAN.3.BP3: Evaluate feasibility of the project. Evaluate the feasibility of achieving the goals of the project in terms of technical feasibility within constraints with respect to time, project estimates, and available resources. [OUTCOME 2]
	MAN.3.BP4: Define, monitor and adjust project activities. Define, monitor and adjust project activities and their dependencies according to defined project life cycle and estimations. Adjust activities and their dependencies as required. [OUTCOME 3, 5, 7] *NOTE 2: A structure and a manageable size of the activities and related work packages support an adequate progress monitoring.* *NOTE 3: Project activities typically cover engineering, management and supporting processes.*
	MAN.3.BP5: Define, monitor and adjust project estimates and resources. Define, monitor and adjust project estimates of effort and resources based on project's goals, project risks, motivation and boundaries. [OUTCOME 2, 3, 7] *NOTE 4: Appropriate estimation methods should be used.*

Fig. 3. MAN.3 process capability levels and base practices for Level 1.

NOTE 5: Examples of necessary resources are people, infrastructure (such as tools, test equipment, communication mechanisms...) and hardware/materials.

NOTE 6: Project risks (using MAN.5) and quality criteria (using SUP.1) may be considered.

NOTE 7: Estimations and resources typically include engineering, management and supporting processes.

MAN.3.BP6: Ensure required skills, knowledge, and experience. Identify the required skills, knowledge, and experience for the project in line with the estimates and make sure the selected individuals and teams either have or acquire these in time. [OUTCOME 3, 7]

NOTE 8: In the case of deviations from required skills and knowledge trainings are typically provided.

MAN.3.BP7: Identify, monitor and adjust project interfaces and agreed commitments. Identify and agree interfaces of the project with other (sub-) projects, organizational units and other affected stakeholders and monitor agreed commitments. [OUTCOME 4, 7]

NOTE 9: Project interfaces relate to engineering, management and supporting processes.

MAN.3.BP8: Define, monitor and adjust project schedule. Allocate resources to activities, and schedule each activity of the whole project. The schedule has to be kept continuously updated during lifetime of the project. [OUTCOME 3, 5, 7]

NOTE 10: This relates to all engineering, management and supporting processes.

MAN.3.BP9: Ensure consistency. Ensure that estimates, skills, activities, schedules, plans, interfaces, and commitments for the project are consistent across affected parties. [OUTCOME 3, 4, 5, 7]

MAN.3.BP10: Review and report progress of the project. Regularly review and report the status of the project and the fulfillment of activities against estimated effort and duration to all affected parties. Prevent recurrence of problems identified. [OUTCOME 6, 7]

NOTE 11: Project reviews may be executed at regular intervals by the management. At the end of a project, a project review contributes to identifying e.g. best practices and lessons learned.

Fig. 3. (*continued*)

Rating Rules: A rating rule [2] provides rating principles which are valid for the majority of assessments. This rule might be sometimes infringed, in this case the assessment report has to include an explanation of the deviation of the rating rule.

Rating Recommendation: A rating recommendation [2] provides a guidance for rating. However, if an assessor decides to not follow the recommendation there is no need to document the deviation in the assessment report.

Traceability and Consistency: Like all previous Automotive SPICE ® assessment model versions ASPICE 3.1 also highlights the importance of a traceability and consistency model. Applying the model in Fig. 4 assures that all requirements are traced throughout the entire development life cycle, e.g. tracing system requirements to system test specifications to system test results (bidirectional).

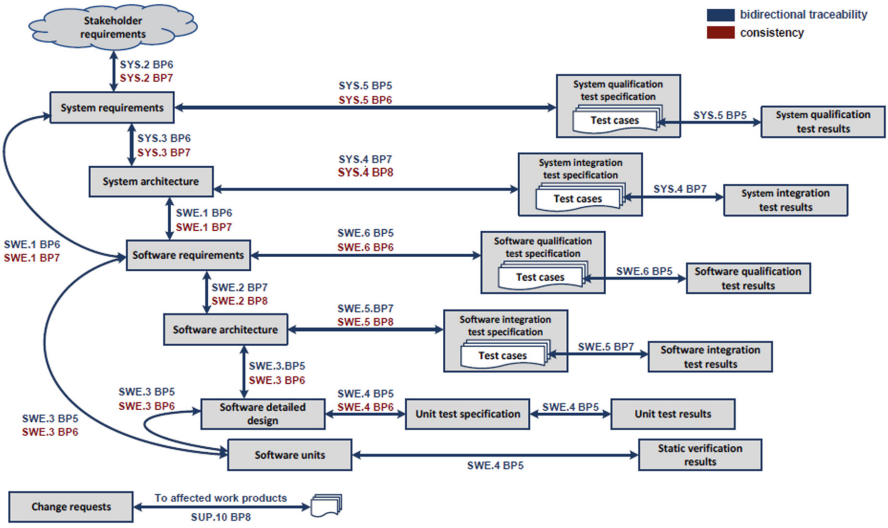

Fig. 4. Traceability and consistency

Once this framework has been set the paper discusses in the next chapter the agile specific rating recommendations in the Automotive SPICE Guideline for the interpretation of Automotive SPICE 3.1.

2 Analysis of the Agile Related Rating Recommendations in the Automotive SPICE Guideline of VDA

The following rating recommendations are described in the VDA Automotive SPICE Guideline.

Planning in Agile Environments

Recommendation AGE.RC.1: If evidences from project planning (e.g. backlog, burndown chart, and/or sprint planning) show gaps regarding release planning and this aspect is significant in the context of MAN.3.BP4, MAN.3.BP9, and SPL.2.BP1, the indicators MAN.3.BP4, MAN.3.BP9, and SPL.2.BP1 should be downrated.

Project Life Cycle

Recommendation AGE.RC.2: If the defined project life cycle does not fit to the project scope, requirements, deliveries, etc. the base practice MAN.3.BP2 should be downrated.

Interpretation of the Recommendations Above

In Automotive projects there is a feature based release plan. Features (vehicle functions) have to be mature (design, tested, released for vehicle testing etc.) at a certain functional integration release. A manufacturer agrees functional integration releases and delivery dates at project start.

Therefore the Change Requests (CR) in the backlog should be grouped into features, features are assigned to integration releases and the planning method for selecting CRs for sprints should consider the planned releases.

In practice there are risks to be managed when applying agile methods in embedded automotive projects. Risk example 1: If the average velocity of sprints (how many CRs per sprint in average are implemented?) is too low the target dates of functional integration releases will not be achieved.

Example 1: In an assessment of an agile automotive project each sprint was reported green/successful. However, in the assessment it was found out that the project was not measuring the size of the backlog. The assessor then measured the size and it contained about 800 CRs which were not considered so far. And the assessor then looked at the average speed of solving CRs in sprints and the full implementation of the backlog would then end 2 years after SOP.

Conclusion to example 1: Even if agile methods are used the release planning in automotive projects still applies.

Risk example 2: If due to the missing velocity of sprints (how many CRs per sprint in average are implemented?) issues remain in the backlog and are not delivered this will be an issue of product liability law if the project is functional safety relevant.

Example 2 additional: In the same project like in example 1 only CRs selected for sprints were analysed, so that the 800 issues in the backlog had not been analysed. A quick review showed that 200 of them were safety relevant [5, 7, 8].

Conclusion to example 2: Even if agile methods are used there must be a release strategy assuring that all relevant features are really implemented. Cars that have not all (e.g. safety relevant or cybersecurity relevant) features [9–16, 26] included are an issue for compliance and the product liability law.

In automotive projects each CR runs through certain phases of the V-model, represented by the SYS and SWE processes in Fig. 1. In Sprints CRs therefore run through the V-Cycle of requirements analysis, architectural design, detailed design, construction, unit testing etc.

Impact of Recommended Rating

If the recommendation AGE.RC.1 is not complete then

- MAN.3.BP4: Define, monitor and adjust project activities.
- MAN.3.BP9: Ensure consistency.
- SPL.2.BP1: Define the functional content of releases. Should be downrated (see Figs. 5 and 6).

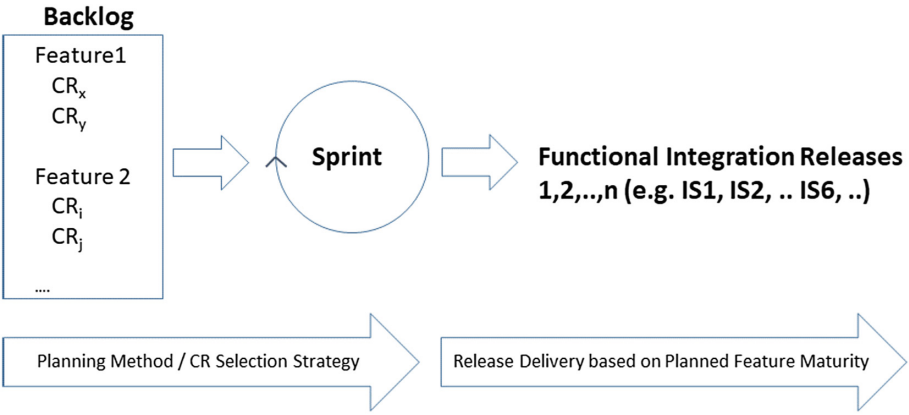

Fig. 5. Planning in agile automotive environments

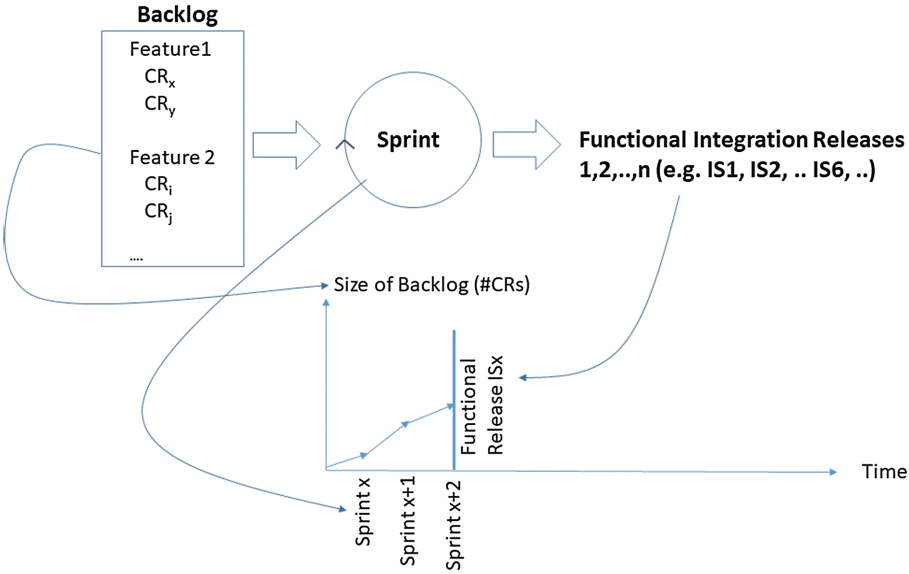

Fig. 6. Consistent sprint and functional release planning

If the recommendation AGE.RC.2 is not complete then

- MAN.3.BP2: Define project life cycle. should be downrated (see Fig. 7).

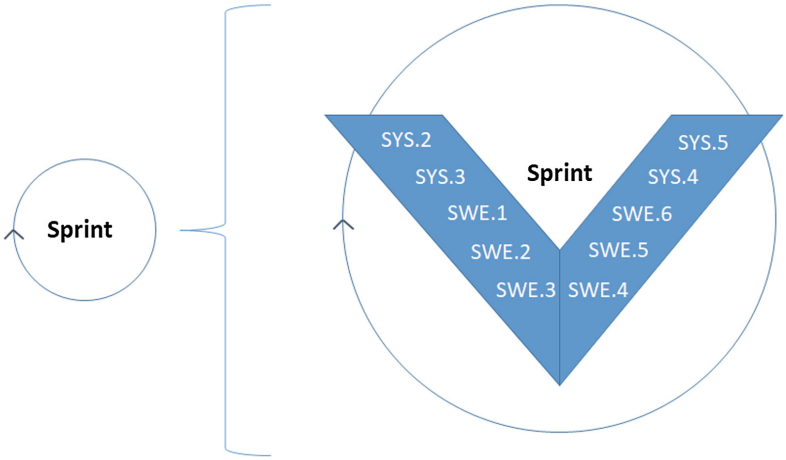

Fig. 7. Phase pattern for agile development and CR tracking

Management of Project Requirements (see Fig. 8**)**

Recommendation AGE.RC.3: If the project development is based on change management without a complete and consistent overview of all project requirements and this aspect is significant in the context of SWE.1.BP3 (for software) and SYS.2.BP3 (for system), then the base practices SWE.1.BP3 (for software) and SYS.2.BP3 (for system) should be downrated.

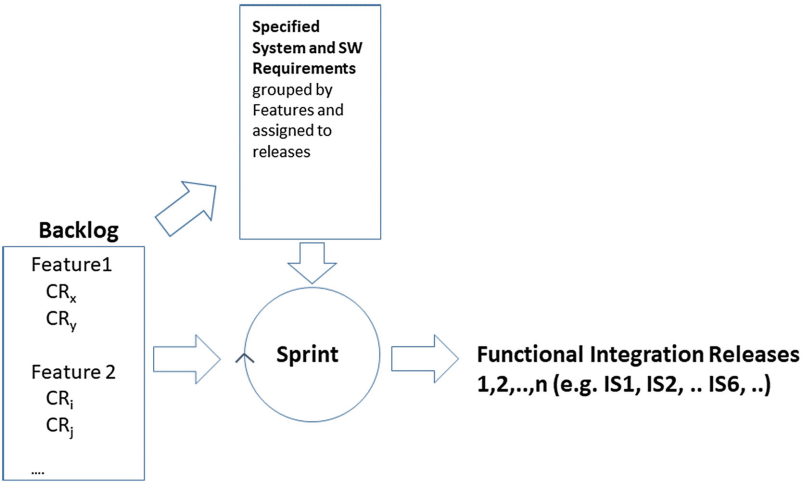

Fig. 8. Agile in automotive needs requirements and not only CRs

Impact of Recommended Rating
If the recommendation AGE.RC.3 is not complete then

- SWE.1.BP3: Analyze software requirements.
- SYS.2.BP3: Analyze system requirements should be downrated (see Fig. 7).

Risk Management

Recommendation AGE.RC.4: If risk management is required for the project but not integrated in the agile project, the base practices MAN.3.BP5 and MAN.3.BP1 should be downrated.

The risk examples 1 and 2 described below Fig. 5 above form typical risks which are observed in agile projects in the automotive embedded systems development. Especially leaving not analysed CRs which might be safety critical and legal issues not analysed in the backlog is a reason to assume that risk management is not properly done.

Testing

Recommendation AGE.RC.7: If the test level software unit verification is not consistently integrated into the agile life cycle, the base practice SWE.4.BP1 should be downrated.

Recommendation AGE.RC.8: If the test level software integration test is not consistently integrated into the agile life cycle, the base practice SWE.5.BP1 (note: there is a mistake in the Automotive SPICE Guideline, it should be SWE.5.BP2) should be downrated.

Recommendation AGE.RC.9: If the test level software qualification test is not consistently integrated into the agile life cycle, the base practice SWE.6.BP1 should be downrated. This means that (see the V inside the sprint in Fig. 7) all different test levels must still be covered.

Impact of Recommended Rating

If the recommendation AGE.RC.7 is not complete then

- SWE.4.BP1: Develop software unit verification strategy including regression strategy should be downrated (see V-Cycle in Fig. 7).

If the recommendation AGE.RC.8 is not complete then

- SWE.5.BP2: Develop software integration test strategy including regression test strategy.should be downrated (see V-Cycle in Fig. 7).

If the recommendation AGE.RC.9 is not complete then

- SWE.6.BP1: Develop software qualification test strategy including regression test strategy.should be downrated (see V-Cycle in Fig. 7).

If one of these above mentioned base practices is downrated then further process specific rating rules apply (not agile specific). For instance, if a test strategy related practice is downrated there are rating rules (not just recommendations) to downrate

further practices. e.g. SWE.4.RL7: If the strategy related activities in SWE.4.BP1 are not performed according to the defined strategy, the indicators SWE.4.BP2 and SWE.4.BP4 shall be downrated.

Software Architecture

Recommendation AGE.RC.5: If no software architecture is developed and maintained, the base practice SWE.4.BP1 should be downrated.

Recommendation AGE.RC.6: If the software architecture is modified/updated incrementally including impact analysis, this should not be used to downrate the indicator SWE.2.BP1 Impact of Recommended Rating: If the recommendation AGE.RC.5 or RC6 is not complete then *SWE.2.BP1: Develop software architectural design should be downrated (see V-Cycle in* Fig. 7*)*. If the base practice SWE.2.BP1 is downrated then further process specific rating rules apply (not agile specific). For instance, if SWE.2.BP1 is doenrated the entire SWE.2 practices on level 1 shall be downrated. e.g. SWE.2.RL.8: If the development of SW architectural design is downrated PA1.1 shall be downrated as all indicators (SWE.2.BP2 to SWE.2.BP9) are affected (Fig. 9).

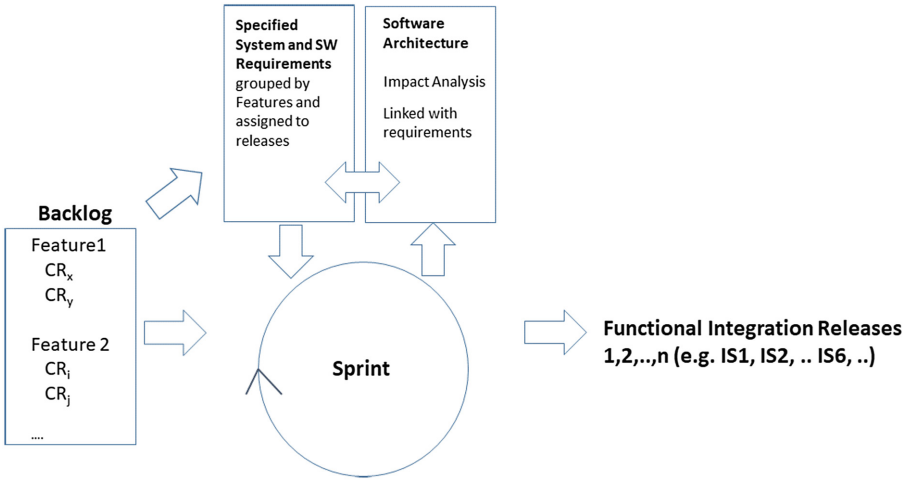

Fig. 9. Agile in automotive needs an incrementally updated SW architecture

Independent Quality Assurance

Recommendation AGE.RC.10: If the project does not ensure that work product and process quality assurance is performed at project level independently and objectively without conflicts of interest, the base practice SUP.1.BP1 should be downrated.

Experience with agile projects in Automotive shows that additional quality metrics are needed. Standard Automotive SPICE ® based metrics (customer requirements coverage, system requirements coverage, software requirements coverage, test

coverage of requirements at all test levels, etc.) have to be enriched by agile based metrics [6].

- Size of the backlog
- Velocity of development (number of CRs per sprint with e.g. monthly sprints)
- Velocity of problem correction (number of CRs of type problem per sprint with e.g. monthly sprints)
- Projected time to solve the backlog
- CRs in different states based om phases (see Fig. 7).

Also the role of quality assurance has still to demonstrate an appropriate level of independence.

3 Expected Impact and Outlook

Automotive SPICE 3.1 ® together with the VDA Automotive SPICE Guideline provides a number of rating recommendations and rating rules that is to be applied by all assessors. There are above 1200 Automotive SPICE assessors active and working in the market. Major manufacturers developed and supported the guide. Therefore a large impact is expected.

The Automotive SPICE Guideline includes a section of how to assess agile development projects in the automotive area. This means on the one hand that agile is accepted but also clearly documents now the requirements and conditions under which agile projects are accepted for embedded automotive systems development.

It is interesting to note that the Automotive SPICE Guideline only considers software architecture, there are no recommendations for systems architecture. However, there are adaptations of the agile manifesto for the systems and product level [20, 21, 27]. This could be included for systems level in future versions of the Automotive SPICE Guideline.

4 The SPI Manifesto Revisited

The SPI manifesto [28–30] describes values and principles which need to be considered to make improvements work in an organisation. One of the approaches is to start with an assessment and to derive an improvement list. This is then used to set up an improvement program [22–25].

The principle "**Use dynamic and adaptable models as needed**" means that depending on the need of organisations specific models can be adapted. The agile manifesto and principles describe such a model.

The principle "**Ensure all parties understand and agree on process**" means that all stakeholders support the improvement project and established processes. An agile approach including sprints and daily stand-ups creates continuous communication in the ream and acceptance by all team members.

An additional principle proposed of the SPI manifesto is: "**Observe new trends and state of the art practices on the market and adopt**". Agile principles became popular in the automotive domain in the last years.

Acknowledgements. We are grateful to iNTACS (www.intacs.info) and VDA for the development of the Automotive SPICE ® 3.1 and the Automotive SPICE Guideline as a guideline to interpret the assessment model.

References

1. Automotive SPICE version 3.1, SPICE User Group, VDA – QMC, November 2017. www.automotivespice.com
2. Automotive SPICE – Guidelines, Process Assessment using Automotive SPICE in the development of software-based systems, 1st edn. VDA – German Automotive Association, Berlin, December 2017. ISSN 0943-9412
3. Manifesto for Agile Software Development (2001). www.agilemanifesto.org
4. Drobietz, M., Poth, A.: A new approach: not agile vs. traditional QM but applying the best of both. In: Stolfa, J., Stolfa, S., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2017. CCIS, vol. 748, pp. 486–494. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64218-5_39
5. ISO - International Organization for Standardization: ISO 26262 Road vehicles Functional Safety Part 1-10 (2011)
6. Kösling, M., Poth, A.: Agile development offers the chance to establish automated quality procedures. In: Stolfa, J., Stolfa, S., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2017. CCIS, vol. 748, pp. 495–503. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64218-5_40
7. SOQRATES: Task Forces Developing Integration of Automotive SPICE, ISO 26262 and SAE J3061. http://soqrates.eurospi.net/
8. Messnarz, R., Kreiner, C., Riel, A.: Integrating automotive SPICE, functional safety, and cybersecurity concepts: a cybersecurity layer model. Soft. Qual Prof. **18**, 13–23 (2016)
9. Messnarz, R., Kreiner, C., Riel, A., et al.: Implementing Functional Safety Standards has an Impact on System and SW Design - Required Knowledge and Competencies (SafEUr), Software Quality Professional (2015)
10. Messnarz, R., Spork, G., Riel, A., Tichkiewitch, S.: Dynamic learning organisations supporting knowledge creation for competitive and integrated product design. In: Proceedings of the 19th CIRP Design Conference – Competitive Design, Cranfield University, 30–31 March 2009, p. 104 (2009)
11. Macher, G., Sporer, H., Brenner, E., Kreiner, C.: Supporting cyber-security based on hardware-software interface definition systems. In: Kreiner, C., O'Connor, R., Poth, A., Messnarz, R. (eds.) EuroSPI 2016. CCIS, vol. 633, pp. 148–159. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-44817-6_12
12. Macher, G., Messnarz, R., Kreiner, C., et al.: Integrated safety and security development in the automotive domain. Working Group 17AE-0252/2017-01-1661, SAE International, June 2017
13. Redmill, F.: Understanding the use, misuse and abuse of safety integrity levels. In: Proceedings of the Eighth Safety-Critical Systems Symposium, Southampton, 8–10 February 2000. Springer, Heidelberg (2000)
14. Much, A.: Automotive security: challenges, standards and solutions. Soft. Qual. Prof. **18**, 4–12 (2016)

15. Messnarz, R., et al.: Integrating functional safety, automotive SPICE and six sigma – the AQUA knowledge base and integration examples. In: Barafort, B., O'Connor, R.V., Poth, A., Messnarz, R. (eds.) EuroSPI 2014. CCIS, vol. 425, pp. 285–295. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43896-1_26
16. Kreiner, C., et al.: Automotive knowledge alliance AQUA – integrating automotive SPICE, six sigma, and functional safety. In: McCaffery, F., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2013. CCIS, vol. 364, pp. 333–344. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39179-8_30
17. Poth, A., Sasabe, S., Mas, A.: Lean and agile software process improvement - an overview and outlook. In: Stolfa, J., Stolfa, S., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2017. CCIS, vol. 748, pp. 471–485. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64218-5_38
18. Poth, A., Wolf, F.: Agile procedures of an automotive OEM – views from different business areas. In: Stolfa, J., Stolfa, S., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2017. CCIS, vol. 748, pp. 513–522. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64218-5_42
19. Siqueira, A.A., Reinehr, S., Malucelli, A.: Using a statistical method to compare agile and waterfall processes performance. In: Stolfa, J., Stolfa, S., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2017. CCIS, vol. 748, pp. 523–532. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64218-5_43
20. Stelzmann, E., Kreiner, C., Spork, G., Messnarz, R., Koenig, F.: Agility meets systems engineering: a catalogue of success factors from industry practice. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 245–256. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15666-3_22
21. Stelzmann, E.S., Spork, G., Kreiner, C.J., Messnarz, R., König, F.: Practical use of agile methods within systems engineering. International Spice Days, June 2010
22. Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.): EuroSPI 2010. CCIS, vol. 99. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15666-3
23. Richardson, I., Runeson, P., Messnarz, R.: Software Process Improvement. LNCS, vol. 4257. Springer, Heidelberg (2006). https://doi.org/10.1007/11908562
24. Barafort, B., O'Connor, R.V., Poth, A., Messnarz, R. (eds.): EuroSPI 2014. CCIS, vol. 425. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43896-1
25. Abrahamsson, P., Baddoo, N., Margaria, T., Messnarz, R. (eds.): EuroSPI 2007. LNCS, vol. 4764. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75381-0
26. Riel, A., et al.: EU project SafEUr – competence requirements for functional safety managers. In: Winkler, D., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2012. CCIS, vol. 301, pp. 253–265. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31199-4_22
27. Wuest, T., Schmidt, T., WeiDavid, W., Romero, D.: Towards (pro-)active intelligent products. Int. J. Prod. Lifecycle Manag. **11**, 154–189 (2018)
28. Korsaa, M., et al.: The SPI manifesto and the ECQA SPI manager certification scheme. J. Soft.: Evol. Process **24**(5), 525–540 (2012)
29. Korsaa, M., et al.: The people aspects in modern process improvement management approaches. J. Soft.: Evol. Process **25**(4), 381–391 (2013)
30. Messnarz, R., Sicilia, M.A., Biro, M.: Social responsibility aspects supporting the success of SPI. J. Soft.: Evol. Process **26**(3), 284–294 (2014)