	1 Define goals, questions and metrics.			4 Identify improvement measures
A: Authors **W:** Workshop	**Goal**	**Questions**	**Metrics**	**Improvement measures**
A	G1: The SW-Requirements [1]Specification (SRS) should be testable.	Q1: What is the complexity of the SRS? Q2: What is the testability of the SRS?	M1.1: Complexity of features and use cases M2.2 Categories, severity and number of anomalies.	I1: Introduce guidelines to assess the complexity of requirements, use cases and business rules. I2: Apply defect taxonomy for requirements validation[2].
A	**G5:** [3]The amount of automated regression test cases should increase from iteration to iteration.	Q1: What is the optimal size [4]of an automated regression test suite?	M1.1: Number and percentage of automated test cases. M1.2: ROI of test automation.	**I1[5]: Take testability of the software artifacts into account, when development starts** I2: Introduce model-based testing.[6]

[1] Feedback of workshop member: Granularity of SRS: Features (high level functionality), requirement, use-case

[2] Felderer M., Beer A.,Using defect taxonomies for requirements validation in industrial projects, RE 2013, IEEE 2013,296-301, 2013.

[3] Goal revised because of feedback of workshop member: “The ROI should be positive for test automation!”

[4] Feedback of workshop member: What is “optimal” depends on the application.

[5] **Feedback of workshop member: This is a good advice supported by research!**

[6] Mohacsi St., Felderer M., Beer A., Estimating the cost and benefit of model-based testing, A decision support procedure for the application of model-based testing in industry, SEAA 2015, Funchal, Portugal, 2015.

A: Authors **W:** Workshop	**Goal**	**Questions**	**Metrics**	**Improvement measures**
W-1	G1: The SW-Requirements [7]Specification (SRS) should be testable.	Q3: Do the requirements provide the necessary information to design test cases? Q4: Are the requirements understandable? Q5: Can you observe the properties of the requirements? Q6: Can I test the red in isolation?	M3.1: Format, completeness (template filled out). M4.1: Requirement smells[8]. M4.2: Review results M4.3: Length of red M6.1: Reference to other requirements.	Requirements-based testing process
W-2	G2: Requirements should be written in a manner that facilitates test-case design	Q1: Can at least one test case be written for each requirement?	M1.1: Number of tests which verifies each requirement (traceability matrix)	I1: Introduce guidelines on language use / restrictions. I2: Use a formalism instead of natural language.
W-3	G3: The SW-requirements should be consistent with each other	Q1: How many contradictions exist between SW-requirements?	M1.1: Number of contradictions	I1. Revise review method
W-4	G4: The SW-requirements should be precise	Q1: How many ambiguous terms exist in the specification? Q2. Are all numeric values qualified with an unit?	M1.1: Number of ambiguous terms M2.1: Number of numeric values that are not qualified with a measurement unit.	I1. Revise review method

[7] Feedback of workshop member: Granularity of SRS: Features (high level functionality), requirement, use-case

[8] Beer A, Junker M., Femmer H.,Felderer M; Initial Investigations on the Influence of Requirement Smells on Test-Case Design; RET 2017.

A: Authors **W:** Workshop	**Goal**	**Questions**	**Metrics**	**Improvement measures**
W-5	G5: The SW-requirements should be atomic.	Q1: How many component requirements specified exist? Q2: How many concepts are covered per requirement?	M1.1: Number of requirements	See also entry W-2.
W-6	G6: Traceability of tests requirements should be maintained	Q1: Can links be maintained between all tests and requirements?	M1.1: Traceability matrix.	I1: Revise test-case design guidelines.
W-7	G7: Number of regression tests.	Q1: Which part of the code has the highest risk (probability and costs)?	M1.1: Code usage in production (per day etc.) M1.2: Revenue per product	I1: Implement measures to comply with business needs? (test coverage?)
W-8	G8: Requirements should be covered by unit test, before going to system test (Effort on unit test coverage impacts on system test).	Q1: What is the coverage of system tests? Q2: What is the result of system test in terms of number of defects found?	M1.1: Number of test cases developed during unit test phase. M2.1: Number of defects found by system test team.	I1: Improve coverage of unit tests.
W-9	G9: Requirements should be updated with the changes in SW-requirements specification.	See also entry W-6.		

Supplement "Goal Question Metrics"

A: Authors **W:** Workshop	**Goal**	**Questions**	**Metrics**	**Improvement measures**
A	G2: The quality of the iterations should continually improve.	Q1: How can we assess the efficiency of test results? Q2: How can we assess the quality of releases?	M1.1: Test results and reports M2.1: History of categories, severity, state and number of defects.	I1: Implement a dashboard to monitor the test process.
A	G3: Change requests should have a low impact on test cases.	Q1: What are the consequences of change requests for testing?	M1.1: Number of regression test cases. M1.2: Added test effort.	I1. Improve maintainability of SW artifacts.
A	G4: The effort of localization, correction, retesting and regression testing of a defect should be minimized.	Q1: How can we assess the testability and observability of requirements?	M1.1: Feature and component dependencies. M1.2: Complexity of the red. M1.3: Effort to localize, fix and retest a defect. M1.4: Number of regression test cases.	I1: Reduce complexity of architecture, requirements and test suite. I2: Improve controllability and observability of SW under test (e.g. via a GUI or API).

Suggestions, Questions?