Software Process Improvement Initiatives based on Quality Assurance Strategies: A QATAM Pilot Application

Dietmar Winkler[1], Frank Elberzhager[2], Stefan Biffl[1], Robert Eschbach[2]

[1] Christian Doppler Laboratory SE-Flex AS,
Institute of Software Technology and Interactive Systems, TU Vienna

[2]Fraunhofer Institute of Experimental Software Engineering (IESE)

dietmar.winkler@tuwien.ac.at

http://cdl.ifs.tuwien.ac.at

Background & Motivation

Common Goals of software development practice:

- High product quality across all phases of software development.
- Optimization of resource allocation and project planning.
- Implementation of process improvement initiatives.

- Process Improvement Initiatives include
  - Application of appropriate software processes according to the project context and application domain.
  - Constructive approaches to create deliverables, e.g., engineering documents, software code, and test cases.
  - Analytical approaches to verify and validate deliverables, e.g., reviews and testing.

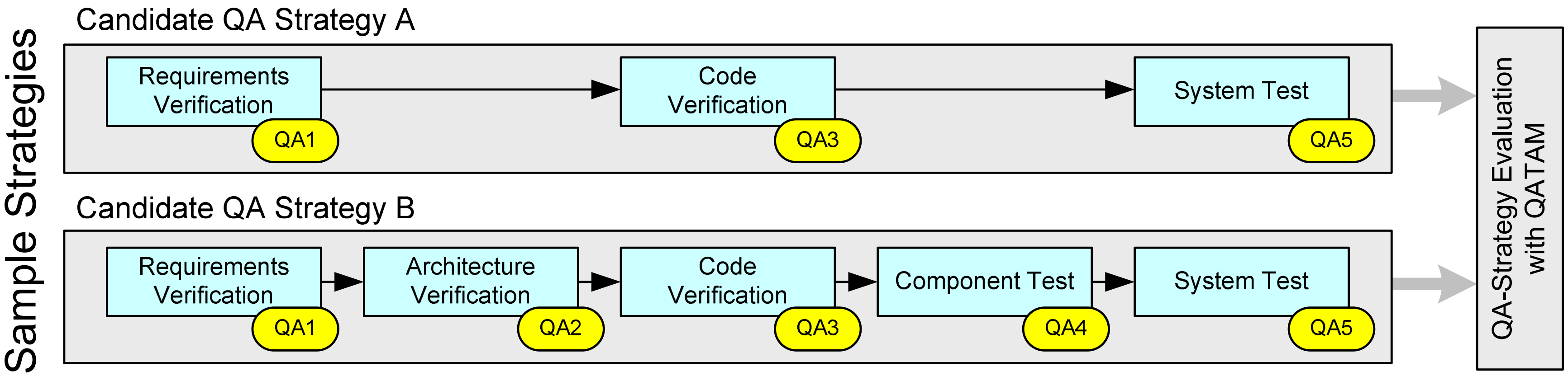
- Quality assurance strategies can help focusing on most promising (bundles of) QA activities.

- QATAM (Quality Assurance Tradeoff Analysis Method) - based on SEIs ATAM - can enable systematic development and evaluation of QA strategies in a given company and project context.

Quality Assurance Strategies

- Bundle of constructive and analytic quality assurance approaches support efficient project planning and execution.

- Observed need for efficient quality assurance strategies:
  - Aligned with the project context and software process.
  - Bundling methods to increase efficiency of project execution (e.g., applying early requirements inspections and derive test cases on acceptance test cases)
  - Ensuring overall high product quality (across all phases of development).
  - Foundation for project planning and resource allocation.



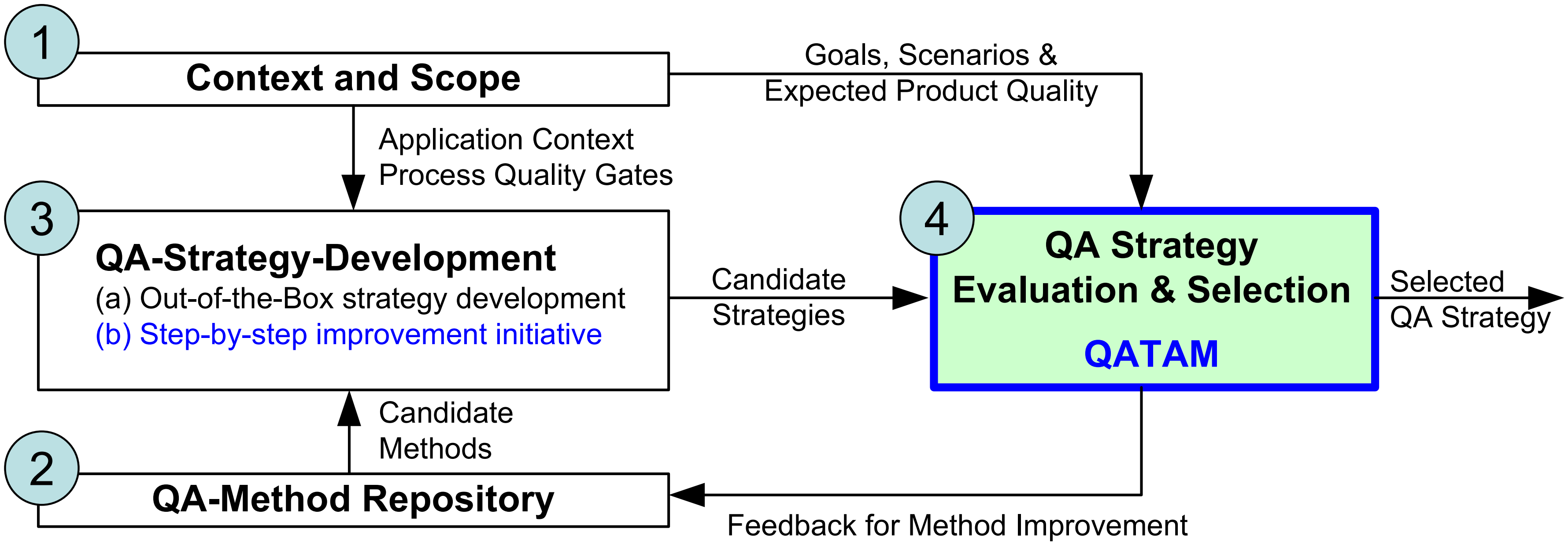
Research Questions & Solution Approach

Research Questions:

- How can we derive an appropriate quality assurance strategy addressing “valuable” project characteristics and quality attributes?
- How can we evaluate quality assurance strategies?

Solution Approach:

- Quality Assurance Tradeoff Analysis Method (QATAM)
- Pilot application in an medium-scale software development company.



QATAM Evaluation Process

- Based on SEI’s Architecture Tradeoff Analysis Method (ATAM) proposed by Kazman *et al*, 1999.
  Qualitative, scenario-based analysis approach to systematically evaluate software architecture variants.

Step	Input	Output
1. QA Strategy Development	Current practices; Involved stakeholders	List of comprehensive Candidate QA strategies
2. Scenario Brainstorming	Context information	Set of scenarios (grouped by stakeholder groups)
3. Pre-selection of possible QA strategies	Set of QA strategies	Refined set of QA strategies
4. Determination of scenario coverage	Refined set of QA strategies & set of grouped scenarios	Estimated scenario coverage regarding each QA strategy
5. Prioritization of scenarios regarding risk and relevance	Set of grouped scenarios	Prioritized set of grouped scenarios
6. Evaluation of QA strategies	Refined set of QA strategies & prioritized set of grouped scenarios	Evaluated QA strategies regarding stakeholder scenario groups and prioritized scenarios
7. Determination of success factors	Refined set of QA strategies & relevant scenario(s)	Analyzed QA strategies regarding determined success factors of relevant scenario(s)
8. Trade-off analysis & determination of one “best-practice” QA strategy.	Refined set of QA strategies, Results of strategy evaluation, results of success factor analysis	One best-practice QA strategy

Pilot Application (1)

- Context is a medium-scale software development organization.
- Focus of improvement: Change Request Handling
- Company need: The improvement initiative has to focus on frequent changing, incomplete and error prone requirements definitions which lead to a high effort in terms of quality, development duration and cost.

- Common goals of the industry partner are

  a) Improvement of software development practices with systematic QA activities.

  b) High stakeholder acceptance of improvement initiatives.

- Step 1: QA Strategy development
  - As-it-is Analysis: simplified V model approach

  Candidate strategies include
  - Method change / extensions, e.g., additional reviews, testing approaches
  - Process change, e.g., V-Modell XT, Scrum application.
  - Strategy development is based on experts/experience and/or empirical evidence.

Pilot Application (2)

- Step 2: Goal/Scenario Brainstorming and Step 5: Prioritization of Goals/Scenarios
  - Based on brainstorming/prioritization processes, e.g., supported by EasyWinWin.
  - Definition of measurement attributes for evaluating strategy performance.
- Step 3: Pre-Selection of candidate QA strategies
  - Determined by company regulations and management decisions.
  - Focus on the 2-3 most promising candidate strategies to limit evaluation effort.
- Step 4: Determination of scenario coverage and Step 6: Evaluation of QA Strategies
  - Workshop for scenario coverage elicitation (% of goal/scenario coverage).
  - Evaluation of goals and scenarios according to (a) risk/priority and (b) stakeholders affected by the goal/scenario.
  - Average coverage of goal/scenario coverage per category.
- Step 7: Determination of success factors
  - Based on success criteria e.g., according to Stelzer et al. 1999.
  - Selected criteria from our industry partner, e.g., strategy performance, effort of implementation, and impact on later stages of development.
- Step 8: Trade-off analysis & determination of one „best-practice“ QA strategy.
  - Based on the evaluation results, an improvement strategy “additional reviews” was selected as first step of an overall improvement initiative.

Summary & Further Work

Summary

- QA strategies enable a comprehensive view on the project by using a set of agreed (bundles of) QA approaches.
- QATAM enables a systematic evaluation according to project and company needs.
- Involving related stakeholders strengthen improvement initiative acceptance.
- "Improvement of small steps" is the most applicable approach in industry context.

Lessons Learned:

- Limiting the number of candidate strategies increase efficiency and effectiveness of strategy evaluation.
- Focus on domain specific QA strategies.
- Application of expert estimation and empirical evidence of methods support (automation-supported) strategy development.

Future work

- Automating the mapping process of project context and method characteristics to generate candidate strategies.
- Refinement of the QATAM evaluation approach based on initial lessons learned.
- Empirical studies on QATAM application in industry context.

Thank you ...

Software Process Improvement Initiatives based on Quality Assurance Strategies: A QATAM Pilot Application

Dietmar Winkler[1], Frank Elberzhager[2], Stefan Biffl[1], Robert Eschbach[2]

[1]Christian Doppler Laboratory SE-Flex AS,
Institute of Software Technology and Interactive Systems, TU Vienna

[2]Fraunhofer Institute of Experimental Software Engineering (IESE)

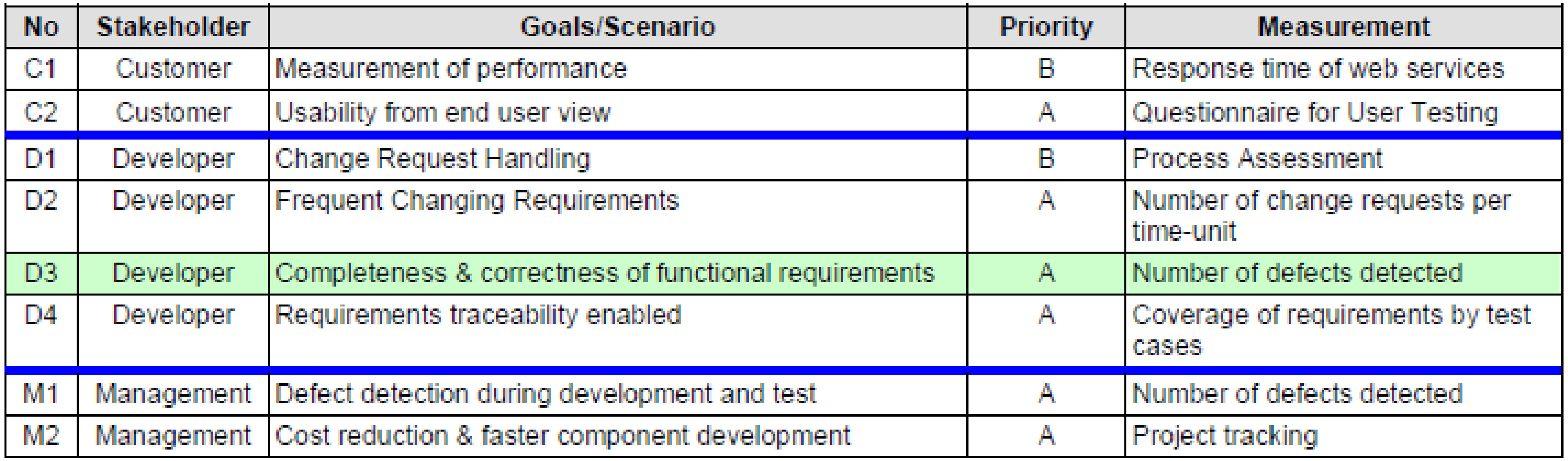
http://cdl.ifs.tuwien.ac.at
Dietmar.Winkler@tuwien.ac.at

Backup Slides

Strategy & Scenario Brainstorming

- Step 1: QA Strategy development
  - As-it-is Analysis (simplified V model approach)
  - Method change / extension, e.g., additional reviews, testing approaches
  - Process change, e.g., V-Modell XT, Scrum application.

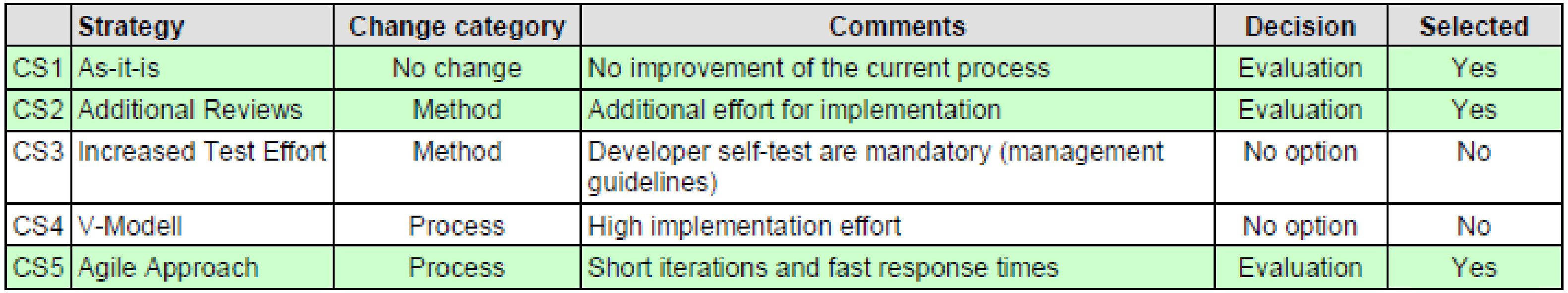
- Step 2: Scenario Brainstorming
  - Goal and Scenario Brainstorming process based on EasyWinWin.
  - Definition of measurement attributes for evaluation purposes.



No	Stakeholder	Goals/Scenario	Priority	Measurement
C1	Customer	Measurement of performance	B	Response time of web services
C2	Customer	Usability from end user view	A	Questionnaire for User Testing
D1	Developer	Change Request Handling	B	Process Assessment
D2	Developer	Frequent Changing Requirements	A	Number of change requests per time-unit
D3	Developer	Completeness & correctness of functional requirements	A	Number of defects detected
D4	Developer	Requirements traceability enabled	A	Coverage of requirements by test cases
M1	Management	Defect detection during development and test	A	Number of defects detected
M2	Management	Cost reduction & faster component development	A	Project tracking

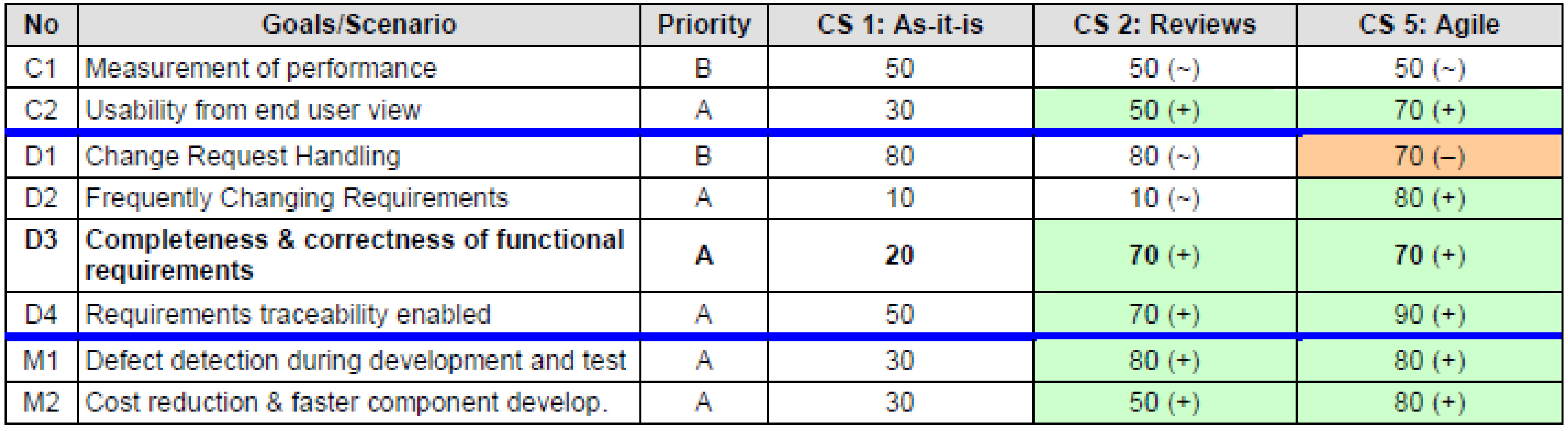
Pre-Selection & Scenario Coverage

- Step 3: Pre-Selection of candidate QA strategies
  - Determined by company regulations and management decisions



	Strategy	Change category	Comments	Decision	Selected
CS1	As-it-is	No change	No improvement of the current process	Evaluation	Yes
CS2	Additional Reviews	Method	Additional effort for implementation	Evaluation	Yes
CS3	Increased Test Effort	Method	Developer self-test are mandatory (management guidelines)	No option	No
CS4	V-Modell	Process	High implementation effort	No option	No
CS5	Agile Approach	Process	Short iterations and fast response times	Evaluation	Yes

- Step 4: Determination of scenario coverage
  - Workshop for coverage determination
  - Involvement of all related stakeholders

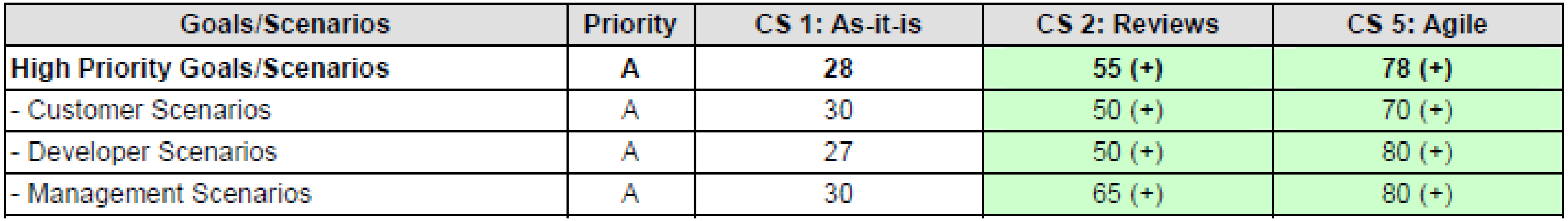


No	Goals/Scenario	Priority	CS 1: As-it-is	CS 2: Reviews	CS 5: Agile
C1	Measurement of performance	B	50	50 (~)	50 (~)
C2	Usability from end user view	A	30	50 (+)	70 (+)
D1	Change Request Handling	B	80	80 (~)	70 (–)
D2	Frequently Changing Requirements	A	10	10 (~)	80 (+)
D3	**Completeness & correctness of functional requirements**	**A**	**20**	**70** (+)	**70** (+)
D4	Requirements traceability enabled	A	50	70 (+)	90 (+)
M1	Defect detection during development and test	A	30	80 (+)	80 (+)
M2	Cost reduction & faster component develop.	A	30	50 (+)	80 (+)

Scenario Prioritization & Evaluation

- Step 5: Prioritization of Scenarios regarding risk and relevance
  - Goal and Scenario Prioritization process based on EasyWinWin.
  - Involvement of related stakeholders.

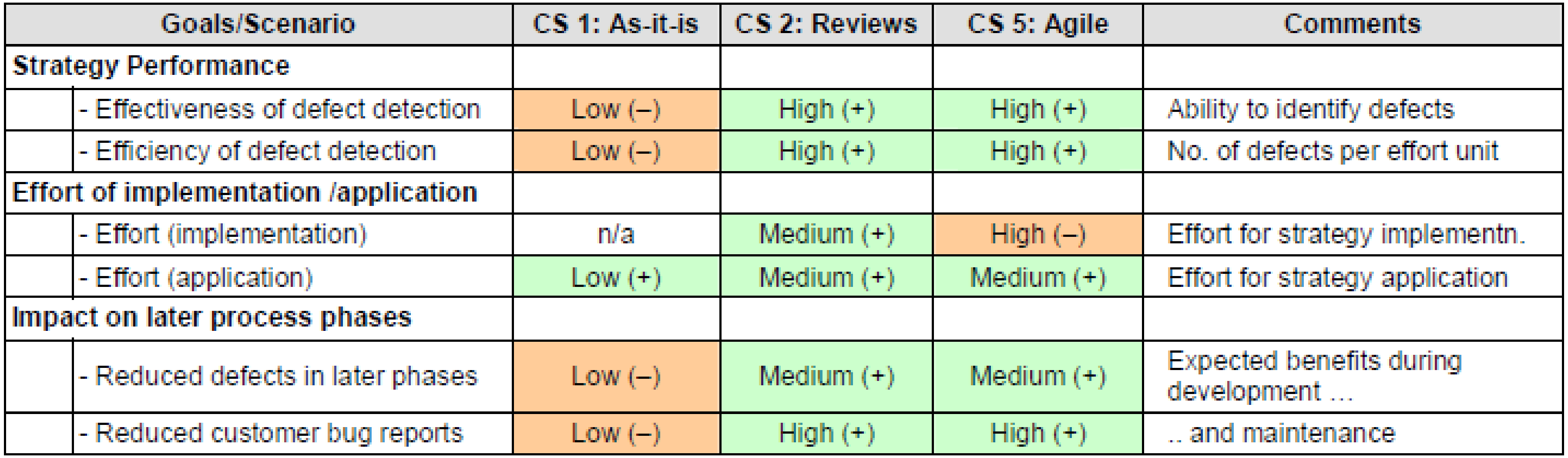
- Step 6: Evaluation of QA Strategies
  - Evaluation of goals and scenarios according to (a) risk/priority and (b) stakeholders affected by the goal/scenario.
  - Mean value per category.



Goals/Scenarios	Priority	CS 1: As-it-is	CS 2: Reviews	CS 5: Agile
High Priority Goals/Scenarios	**A**	**28**	**55 (+)**	**78 (+)**
- Customer Scenarios	A	30	50 (+)	70 (+)
- Developer Scenarios	A	27	50 (+)	80 (+)
- Management Scenarios	A	30	65 (+)	80 (+)

Success Factors & Final Decision

- Step 7: Determination of success factors
  - Based on success criteria according to Stelzer *et al.* 1999.
  - Selected criteria, e.g., strategy performance, effort of implementation, and impact on later stages of development.



Goals/Scenario	CS 1: As-it-is	CS 2: Reviews	CS 5: Agile	Comments
Strategy Performance				
- Effectiveness of defect detection	Low (–)	High (+)	High (+)	Ability to identify defects
- Efficiency of defect detection	Low (–)	High (+)	High (+)	No. of defects per effort unit
Effort of implementation /application				
- Effort (implementation)	n/a	Medium (+)	High (–)	Effort for strategy implementn.
- Effort (application)	Low (+)	Medium (+)	Medium (+)	Effort for strategy application
Impact on later process phases				
- Reduced defects in later phases	Low (–)	Medium (+)	Medium (+)	Expected benefits during development …
- Reduced customer bug reports	Low (–)	High (+)	High (+)	.. and maintenance

- Step 8: Trade-off analysis & determination of one „best-practice“ QA strategy.
  - Based on the results, the Candidate Strategy 2 (additional reviews) was selected as most valuable strategy for the first step of the improvement initiative.