

QATAM: ein Szenario-basierter Ansatz zur Evaluierung von Qualitätssicherungsstrategien

Dietmar Winkler^{1,3}, Christian Denger², Frank Elberzhager², Stefan Biffel^{1,3}

¹Institute of Software Technology and Interactive Systems,
Vienna University of Technology

²Fraunhofer Institute of Experimental Software Engineering (IESE)

³OCG Arbeitskreis „Software Prozesse“

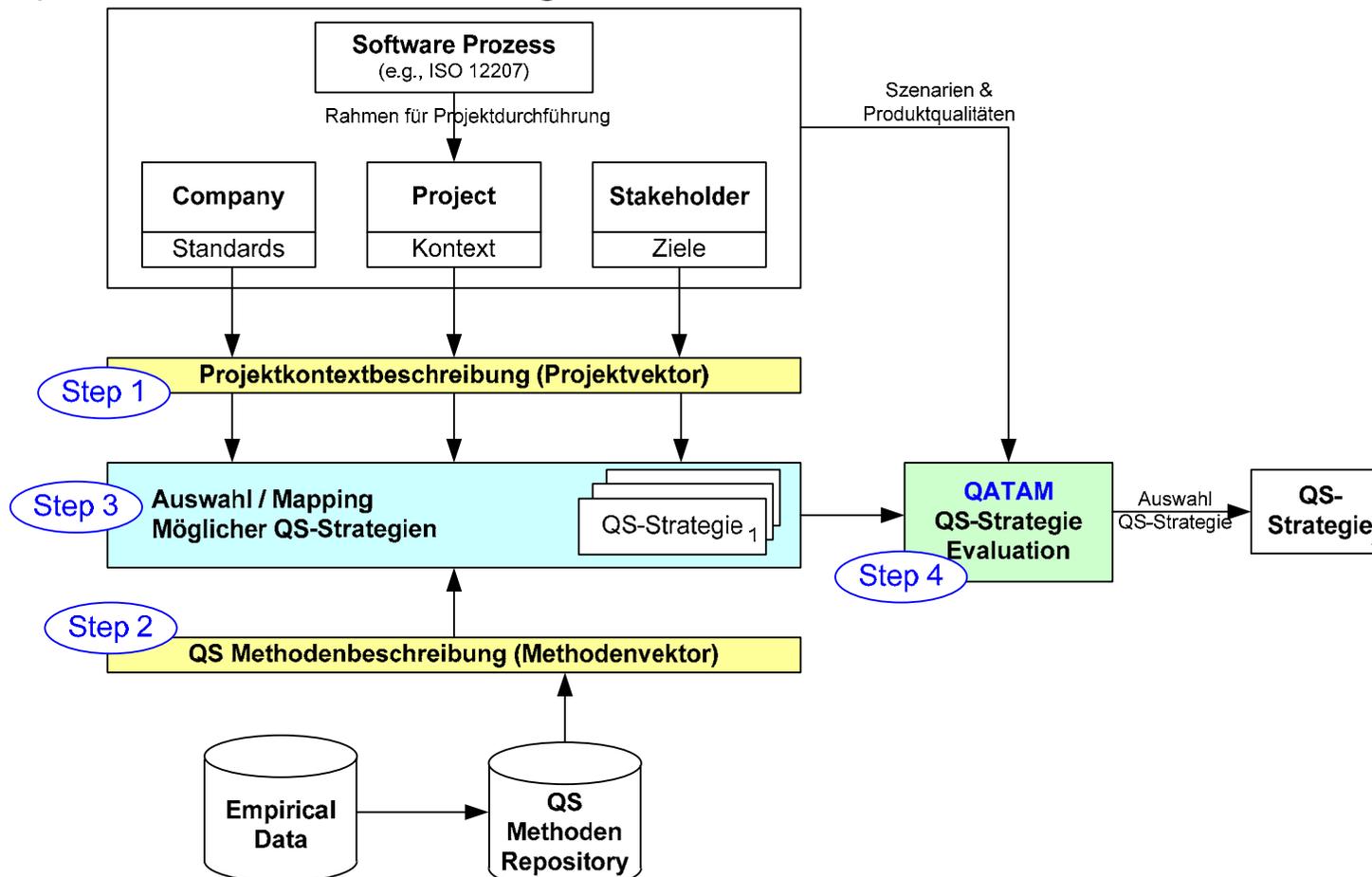
dietmar.winkler@qse.ifs.tuwien.ac.at
<http://qse.ifs.tuwien.ac.at>

- Ziel moderner Software- und Systementwicklung ist die Herstellung **qualitativ hochwertiger Produkte** innerhalb von Zeit- und Kostenvorgaben.
- Effiziente Softwareentwicklung erfordert eine **angemessene Projektplanung** für die Herstellung und die Evaluierung der Produkte.
- Eine systematische Planung umfasst:
 - Festlegung eines **geeigneten Vorgehensmodells**, wie z.B. V-Modell XT.
 - **Konstruktive Methoden** zur Herstellung der Produkte in unterschiedlichen Phasen der Entwicklung.
 - **Analytische Methoden** zur Verifikation und Validierung von Produkten und zur Einschätzung der Produktqualität, wie z.B. Inspektion und Testen.
- Ziele:
 - Einsatz aufeinander **abgestimmter Methoden** zur Erreichung einer **durchgängig hohen Produktqualität** in allen Phasen der Entwicklung sowie eine **Optimierung** der Ressourcen.
 - Verbesserung des Entwicklungsprozesses durch Qualitätsstrategien.

- In der gängigen Praxis werden Methoden zur Qualitätsverbesserung häufig punktuell eingesetzt, daher:
 - **punktuelle Produktverbesserungen**, z.B. Einsatz von Inspektionen
→ keine durchgängige Sicht auf die Produktqualität und **fehlende Gesamtsicht** auf das Projekt
 - Meist **redundante Aktivitäten** und **ineffiziente Ressourcenplanung**, z.B. Inspektion zur Fehlerfindung und Erstellung von Testfällen in unterschiedlichen Arbeitsschritten.
 - **Trennung der Rollen** (unterschiedliche Abteilungen), z.B. QS und Tester.
 - Einsatz von „**bewährten**“ **Methoden** anstatt dem Einsatz von „Best-Practice Methoden“.
 - **Voraussetzungen für Methodeneinsatz nicht erfüllt**, z.B. keine finale Version Produktversion → Reduzierte Wirksamkeit der QS-Methode.
 - Fokus auf „untergeordnete“ Produktqualitäten.
- Der Einsatz einer Qualitätsstrategie ermöglicht eine effizientere Planung, eine ganzheitliche Sicht auf das Projekt und die Produktqualität.

- Eine QS-Strategie umfasst ein **Bündel von konstruktiven und analytischen Maßnahmen** zur effizienteren Projektabwicklung.
- Vorteile:
 - Abgestimmt auf **Projektkontext** und den **Software Prozess**.
 - Nutzung von **Synergieeffekte** durch die **Kombination von Methoden** (z.B. Inspektion von Anforderungen und Ableitung von Testfällen für Akzeptanz- oder Abnahmetests).
 - Sicherstellung einer durchgängigen **Produktqualität**.
 - Grundlage zur Verbesserung der **Projektplanung** und **Optimierung der Ressourcenauslastung**.
- Zentrale Fragestellungen:
 - Wie können QS-Maßnahmen auf das Projekt bzw. aufeinander **abgestimmt** werden?
 - Wie kann eine QS-Strategie „**wertvolle**“ **Produktqualitäten adressieren**?
 - Wie können Qualitätsstrategien **evaluiert** werden?

- Voraussetzung für die Auswahl der Qualitätssicherungsstrategie sind:
 - a) Beschreibungen des **Projektkontextes**.
 - b) Beschreibung von **Methoden**.
 - c) Beschreibung von **Szenarien und Produktqualitäten**.
 - d) effiziente Evaluierungsschemata.



- Wie können QS-Maßnahmen auf das Projekt bzw. aufeinander abgestimmt werden?
- Step 1: Beschreibung von **Kontextfaktoren des aktuellen Projektes**, z.B.
 - Verwendeter Software Prozess (z.B. Life-Cycle, V-Modell XT) und Unternehmensstandards (z.B. Company Best-Practices, Strategie)
 - Projektkontext (z.B. Dauer, Komplexität, Risiko, Domäne, Ressourcen)
 - Stakeholder (z.B. Funktionale / Nicht-Funktionale Anforderungen)
- Step 2: **Methodenbeschreibungen**, z.B.
 - Erforderliche Erfahrung der Anwender
 - Anwendungsbereich, adressierte Qualitätsattribute, Individuell erwarteter Nutzen
 - Aufwand und Kosten, ergänzende Methoden
- Step 3: **Mapping** von Projektkontextbeschreibung und Methodenbeschreibungen (Step 1-2) → Mögliche QS-Strategien.
- Step 4: **Evaluierung** der erhaltenen möglichen Strategien.

- Wie kann eine QS-Strategie „wertvolle“ Produktqualitäten adressieren?
- Durch die Ermittlung und Priorisierung von Szenarien und Produktqualitäten aller involvierten Stakeholder.
 - Szenarien, z.B. häufig geänderte Anforderungen
 - Produktqualitäten, z.B. Benutzerfreundlichkeit der Anwendung
- Priorisierung von Szenarien und Produktqualitäten unter Einbindung aller involvierten Stakeholder.
- Werkzeugunterstützung durch den EasyWinWin Prozess (Boehm et al, 2001)
- Grundlegende Vorgehensweise:
 - Identifikation aller Stakeholder Win-Conditions.
 - Identifikation von Konflikten (*conflicts*) und Problempunkte (*issues*).
 - Erarbeiten von alternativen Lösungen (*options*).
 - Ungelöste Problempunkte werden definierte Risiken.
 - Ziel: Zustimmung (*agreement*) aller Stakeholder.

- Wie können Qualitätsstrategien evaluiert werden?
- QATAM: Quality Assurance Tradeoff Analysis Method
 - Basiert auf SEI's ATAM (*architecture tradeoff analysis method*) zur Evaluierung unterschiedlicher Architekturvarianten (Kazman et al, 1999).
 - ATAM ist eine qualitative, szenariobasierte Analysetechnik zur strukturierten Analyse einer Architektur.
 - Der Evaluierungsprozess umfasst - in Anlehnung an ATAM - 9 Schritte:
 1. Planung des QS-Strategiefindungsprozesses
 2. Szenarien Brainstorming
 3. Initiale Vor-Auswahl von möglichen QS-Strategien
 4. Ermittlung des Erfüllungsgrades (Scenario Coverage) der QS-Strategien
 5. Priorisierung von Szenarios nach Risiko und Relevanz (z.B. A = kritisch, B = wichtig und C = unkritisch)
 6. Evaluierung der QS-Strategie (z.B. Fokus auf kritische Szenarien)
 7. Ermittlung der Erfolgsfaktoren
 8. Tradeoff-Abschätzung
 9. Ermittlung der „Best-Practice-Strategie“ und Projektplanung

- Problemstellung:
 - Anforderungen sind häufig unvollständig bzw. fehlerhaft und führen meist zu erheblichem Mehraufwand (Zeit und Kosten).
- Zielsetzung:
 - Verbesserung der Softwareentwicklung durch systematischen Einsatz von Qualitätssicherungsmaßnahmen.
 - Schwerpunkt: Change Request Handling.
 - Geringer „Reibungswiderstand“ der beteiligten Stakeholder.
- Strategische Umsetzung:
 - Ermittlung von wichtigen, zu adressierenden Szenarien.
 - Erhebung des Ist-Zustandes.
 - Erarbeitung unterschiedlicher Qualitätssicherungsstrategien.
 - Initiale Vorauswahl von möglichen Strategien (Unternehmensvorgabe).
 - Evaluierung durch QATAM.

- Auszug der Ergebnisse aus einem Ziel- und Szenario-Brainstorming Prozess (Schritt 2) sowie Priorisierung (Schritt 5):

Customer

Nr.	Szenarien	Priorität
C1	Performancemessungen (Response time of web-services).	B
C2	Benutzerfreundlichkeit aus Sicht des Endbenutzers.	A

Entwickler / Developer

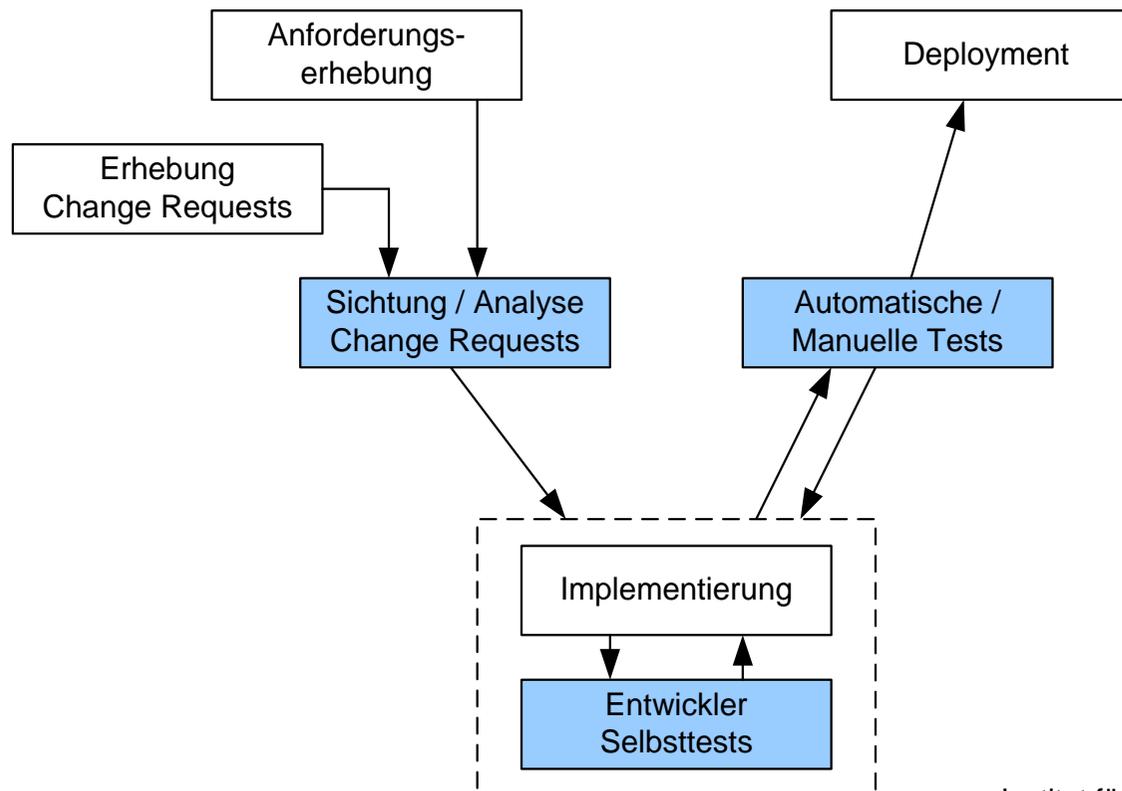
Nr.	Szenarien	Priorität
D1	Change Request Handling.	B
D2	Häufig geänderte Anforderungen.	A
D3	Vollständigkeit und Richtigkeit der funktionalen Anforderungen.	A
D4	Testbarkeit der Anforderungen.	A

Management

Nr.	Szenarien	Priorität
M1	Fehlererkennung während Entwicklung und Tests.	A
M2	Kosteneinsparungen und schnellere Verfügbarkeit von Komponenten.	A
M3	Kosten für die Einführung.	A

Strategie S1: Ist-Prozess

- Auszug aus der Ist-Analyse:
 - Der Entwicklungsprozess orientiert sich weitgehend am V-Modell.
 - Sichtung und Klassifikation der Change Requests.
 - Entwicklerselbsttest während der Codierungsphase.
 - Automatische und manuelle Tests, Releasetests.



Strategie S2 (Zusätzliche Reviews):

- Geringfügige Anpassung des Entwicklungsprozesses durch eine zusätzliche Reviewphase.
- Requirements-Reviews z.B. im Rahmen der Sichtung und Klassifikation von Change Requests, um Anforderungen frühzeitig auf Richtigkeit zu prüfen.

Strategie S3 (Anpassung des Testaufwandes):

- Erhöhen des Testaufwandes um etwa 20% bei automatisierten Tests, bei gleichzeitiger Reduktion der Entwicklerselbsttest.

Strategie S4 (V-Modell):

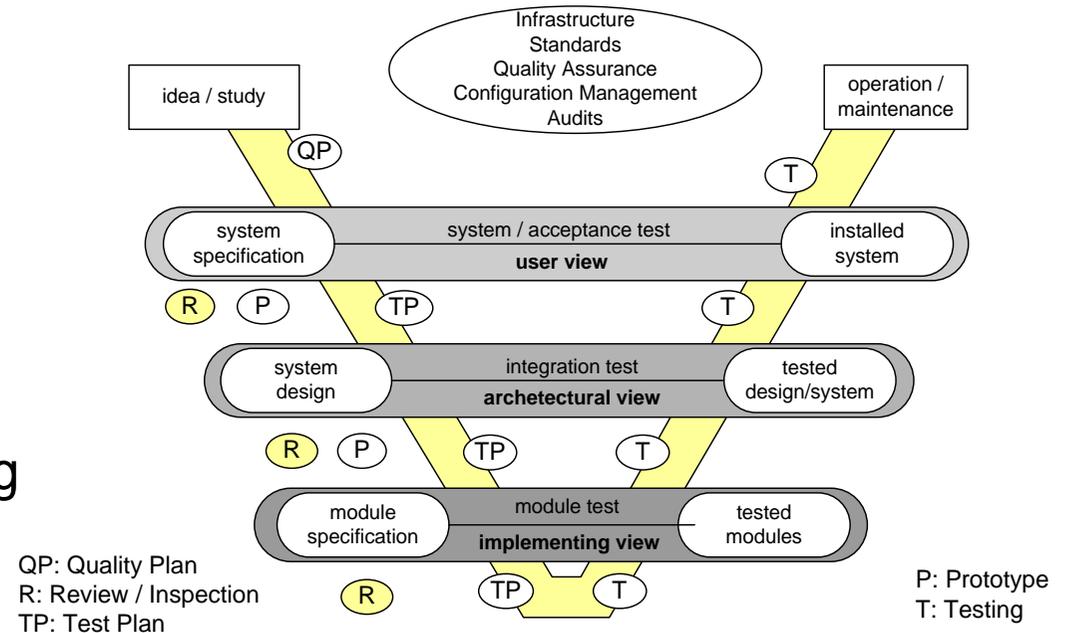
- Anpassung des Entwicklungsprozesses auf das V-Modell mit integrierter Qualitätssicherung.

Strategie S5 (Agiler Ansatz):

- Unterstützung von kurzen Iterationen und sich häufig ändernden Anforderungen durch SCRUM und Test-Driven Development.

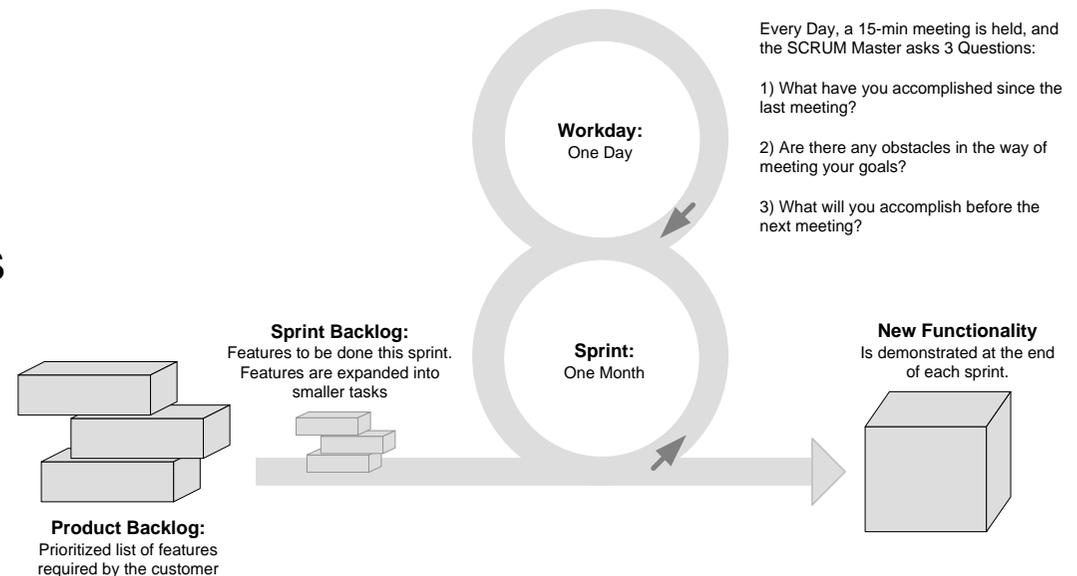
Strategie S4 (V-Modell):

- Einsatz umfassender und systematischer QS-Methoden
- Reviews und Inspektionen
- Testplanung und -durchführung



Strategie S5 (Agiler Ansatz):

- Flexible Entwicklung auf SCRUM Basis
- Test-Driven Development als agile Praxis



- Schlüsselszenario: „Überprüfung auf Vollständigkeit und Richtigkeit der funktionalen Anforderungen“ (D3)
- Initiale Vorauswahl der Strategie (Schritt 3) durch die gemeinsame Auswahl aller involvierten Stakeholder.

Nr.	Strategie	Entscheidung	Begründung
S1	Ist-Prozess	Evaluierung	Keine Veränderung des bisherigen Prozesses
S2	Zusätzliche Reviews	Evaluierung	Prozessänderung: Mehraufwand am Beginn der Entwicklung
S3	Testaufwand	Nein	Prozessänderung: Entwicklertests sind Managementvorgabe
S4	V-Modell	Nein	Prozess Change, wird nicht angestrebt
S5	Agil	Evaluierung	Prozess Change: Kurze Iterationen und schnelle Reaktionsmöglichkeit

- Die Strategien S1 (Ist-Prozess), S2 (Zusätzliche Reviews) und S5 (Agiler Ansatz) werden weiter verfolgt und evaluiert.

- Erfüllungsgrad (0-100%) der Szenarien durch den Einsatz der Strategien (Schritt 4).
- Schätzwerte durch
 - Expertenschätzungen
 - Empirische Daten (z.B. Projektdaten)
- Ergebnis: Strategie 5 (agile) ist zu bevorzugen.

Szenario	Priorität	Strategien		
		S1: Ist-Prozess	S2: Reviews	S5: Agile
C1 Performance measures	B	50	50	50
C2 Application Usability	A	30	50	70
D1 Change Request Handling	B	80	80	70
D2 Requirements Changes	A	10	10	80
D3 Completeness of Requirements	A	20	70	70
D4 Testability of requirements	A	50	70	90
M1 Defect detection during development	A	30	80	80
M2 Cost reduction / faster delivery	A	30	50	80

- Evaluierung der Strategien (Schritt 6)
- Gruppiertes Mittelwert über Szenario Prioritäten und Stakeholder Gruppen
- S2 und S5 bringen deutliche Verbesserungen für kritische Szenarien („high-priority“), wobei die „agile“ Strategie vorzuziehen ist.

Szenario	Priorität	Strategien		
		S1: Ist-Prozess	S2: Reviews	S5: Agile
High Priority Scenarios	A	28	55	78
- Customer Scenarios	A	30	50	70
- Developer Scenarios	A	27	50	80
- Management Scenarios	A	30	65	80
Important Scenarios	B	65	65	60
- Customer Scenarios	B	50	50	50
- Development Scenarios	B	80	80	70
- Management Scenarios	B	-	-	-

- Ermittlung der Erfolgsfaktoren (Schritt 7)
- Fokus auf D3: Completeness and Correctness of Functional Requirements
- Schätzwerte durch
 - Expertenschätzungen
 - Empirische Daten (z.B. Projektdaten)
- Aufgrund der sehr hohen Kosten für eine Prozessänderung wurde die Strategie S2 „zusätzliche Reviews“ bevorzugt (Schritt 8, Tradeoff-Abschätzung basierend auf den Evaluierungsergebnissen)

Szenario	Priorität	Strategien		
		S1: Ist-Prozess	S2: Reviews	S5: Agile
D3 Completeness of Requirements	A	20	70	70
- Effectiveness of defect detection		Low	High	High
- Efficiency of defect detection		Low	High	High
- Effort (implementation)		n/a	Medium	High
- Effort (technique application)		Low	Medium	Medium
- Reduced defects in later phases		Low	Medium	Medium
- Reduced Customer bug reports (field)		Low	High	High

Zusammenfassung

- Der punktuelle Einsatz von QS-Methoden führt zwar zur punktuellen Verbesserung von einzelnen Produkten, birgt aber Risiken wie
 - eine fehlende Gesamtsicht auf das Projekt
 - reduzierte Wirksamkeit des Methodeneinsatz bei Nichterfüllung der Produktvoraussetzungen und
 - ineffiziente Ressourcenplanung durch redundante Aktivitäten
- QS-Strategien ermöglichen den durchgängigen Einsatz von abgestimmten Methoden zur Produkt- und Prozessverbesserung.
- Durch die Einbeziehung der relevanten Stakeholder helfen Szenarien bei der Identifikation und Priorisierung von Schlüsselfaktoren zur Auswahl der Strategie
- Die Evaluierung von Strategien ermöglicht Ansätze zur Produkt- und Prozessverbesserung.

Ausblick

- Automatisierung des Mapping-Prozesses von Projektkontext- und Methodenbeschreibung zur Generierung möglicher QS-Strategien.
- Evaluierung und Verfeinerung des QATAM Prozesses und systematische Untersuchung anhand mehrerer Fallbeispiele.

- Biffi S., Denger C., Elberzhager F., Winkler D.: „*A Quality Assurance Strategy Tradeoff Analysis Method*“, Euromicro Software Engineering and Advanced Applications (SEAA), Lübeck, Germany, 2007.
- Biffi S., Denger C., Elberzhager F., Winkler D. (2007); „*Quality Assurance Tradeoff Analysis Method (QATAM) - An Empirical Quality Assurance Planning and Evaluation Framework*“, Technische Universität Wien, Technical Report IFS-QSE-07/04.
- Boehm B., Grünbacher P., Briggs R.: „*EasyWinWin: A Groupware-Supported Methodology for Requirements Negotiation*“, Int. Conf. on Software Engineering, 2001.
- Denger C., Elberzhager F.: „*Basic Concepts to Define a Customized Quality Assurance Strategy*“, IESE Report No. 013.07, 2007.
- Elberzhager F., Denger C.: „*A Comprehensive Planning Framework for Selecting and Customizing Quality Assurance Techniques*“, Software Engineering and Advanced Applications (SEAA), Lübeck, Germany, 2007.
- Kazman R., Barbacci M., Klein M., Carriere S.J., Woods S.G.: „*Experiences with Performing Architecture Tradeoff Analysis*“, Int. Conf. on Software Engineering, 1999.
- Schwaber K.: „*Agile Project Management with Scrum*“, Microsoft Press, 2004.