

# Projektbezogene Auswahl von Bewertungsmethoden für Softwarearchitekturen

Prof. Dr. Stefan Eicker (Universität Duisburg-Essen), Dipl.-Wirt.-Inf. Christian Hegmanns (Accenture GmbH), Dr. Stefan Malich (Universität Duisburg-Essen)

Lehrstuhl für Wirtschaftsinformatik und Softwaretechnik  
Universität Duisburg-Essen (Campus Essen)  
Universitätstrasse 9  
45127 Essen

stefan.eicker@icb.uni-due.de  
christian.hegmanns@accenture.de  
stefan.malich@icb.uni-due.de

**Abstract:** Die Softwarearchitektur eines Anwendungssystems besitzt einen erheblichen Einfluss auf die Qualitätsmerkmale des Gesamtsystems. Sie stellt somit einen kritischen Erfolgsfaktor sowohl für die Entwicklungs- als auch für die Nutzungsphase des Systems dar. In einem Softwareentwicklungsprojekt sollte deshalb eine dedizierte und detaillierte Betrachtung der Softwarearchitektur erfolgen. Zur Bewertung einer Softwarearchitektur im Kontext eines konkreten Projektes ist eine Vielzahl von Bewertungsmethoden mit unterschiedlichen Eigenschaften entwickelt worden. Der vorliegende Beitrag beschäftigt sich mit der Auswahl einer geeigneten Methode für ein Projekt; er stellt dazu eine Taxonomie und darauf aufbauend einen strukturierten Auswahlprozess vor.

## 1 Einleitung

Eine Softwarearchitektur beschreibt die wesentlichen Softwareelemente eines Anwendungssystems, die Eigenschaften der Elemente und die Beziehungen zwischen ihnen (vgl. [BCK03, S. 21; SG96, S. 1]). Sie ist die Grundlage für die Aufwandschätzung und Projektplanung sowie ein wichtiges Kommunikationsinstrument, insbesondere bei der Entwicklung in großen Teams.

Im Kontext der Entwicklungsarbeiten definiert die Softwarearchitektur einen übergeordneten Rahmen, der die Grundlage für alle weiteren Entwürfe des Systems bildet. Einen entsprechend großen Einfluss besitzen die Entscheidungen, die für ihre Struktur getroffen werden, auf die nicht-funktionalen Qualitätsmerkmale des Anwendungssystems: Sie unterstützen oder behindern die Erreichung nahezu aller Merkmale (vgl. [CKK02, S. 20]). Die Bewertung einer Softwarearchitektur bildet somit ein wichtiges Instrument des Risikomanagements, da sie die frühzeitige Identifikation von Abweichungen in Bezug auf die Erreichung der Qualitätsanforderungen und in Bezug auf die Zeit- und Budgetplanung ermöglicht (vgl. [CKK02, S. 23-24]).

Um Softwarearchitekturen bewerten zu können, müssen die Auswirkungen der zugehörigen Entwurfsentscheidungen auf die Qualitätsmerkmale *systematisch* analysiert werden. Der Aufwand für die Bewertung einer konkreten Architektur ist dann relativ gering und es ergibt sich ein sehr gutes Kosten-/Nutzen-Verhältnis (vgl. [CKK02, S. 23-24, 39 ff.]). Trotz der hohen Relevanz und des guten Kosten-/Nutzen-Verhältnisses der Bewertung einer Softwarearchitektur ist sie dennoch in der Praxis häufig *kein* fester Bestandteil des Softwareentwicklungsprozesses. Dies resultiert nicht zuletzt daraus, dass zwar eine relativ große Anzahl von Bewertungsmethoden entwickelt worden ist, dass aber mit sehr unterschiedlichen Anforderungen, Schwerpunkten und Zielen. Welche Methode in einem konkreten Fall genutzt werden sollte bleibt offen, d. h. eine *strukturierte Unterstützung* für den Auswahlprozess existiert bisher nicht.

Als Grundlage für die Entwicklung eines Auswahlverfahrens stehen strukturierte Vergleiche der existierenden Bewertungsmethoden zur Verfügung: *Clements*, *Kazman* und *Klein* haben einen Vergleich auf Basis von fünf Kriterien durchgeführt (siehe [CKK02, S. 255 ff.]). Sie beziehen aber nur am Software Engineering Institute (SEI)<sup>1</sup> entwickelte Bewertungsmethoden<sup>2</sup> ein. Außerdem wird die Auswahl einer Methode nicht durch einen strukturierten Prozess unterstützt.

*Dobrica* und *Niemelä* haben eine Klassifizierung entwickelt und acht Bewertungsmethoden einander gegenübergestellt (siehe [DN02]). Die Klassifizierung gibt einige Richtlinien an, die den Vergleich der Methoden im Kontext der Bewertung einer Architektur unterstützen sollen (vgl. [DN02, S. 648]). Insgesamt wurden sieben Kriterien identifiziert, wobei allerdings nur vier Kriterien zur Klassifizierung der Bewertungsmethoden verwendet werden (Evaluationstechnik, Qualitätsmerkmal, Stakeholder-Beteiligung und die Phase, in der die Methode angewendet werden kann) (vgl. [DN02, S. 648-649]). Kritisch ist anzumerken, dass sich die Kriterien und Richtlinien ausschließlich auf den Vergleich beziehen und die eigentlichen Auswahlsschritte nicht im Detail diskutiert werden.

*Babar*, *Zhu* und *Jeffrey* haben auf der Basis der zuvor genannten Arbeiten einen Vergleich von acht Bewertungsmethoden durchgeführt, wobei sie für den Vergleich fünfzehn Kriterien verwendet haben (siehe [BZJ04]). Ziel des Vergleichs war eine ausführliche Beschreibung der Merkmale einer Methode sowie die Erfassung der Aspekte einer Methode, die im Allgemeinen bekannt, jedoch nur unzureichend oder informell beschrieben worden sind, so dass primär die Gemeinsamkeiten und Unterschiede der Methoden identifiziert werden können (vgl. [BZJ04, S. 309-310]). Mit Ausnahme der szenariospezifischen Kriterien sind alle Kriterien von *Dobrica* und *Niemelä* auch in diesem Vergleich zu finden. Darüber hinaus verwenden *Babar*, *Zhu* und *Jeffrey* zehn weitere Kriterien, wie z. B. das Ziel und die Prozessunterstützung der Methode (vgl. [BZJ04, S. 310 ff.]). Analog zu den Arbeiten von *Dobrica* und *Niemelä* beschränken sich *Babar*, *Zhu* und *Jeffrey* auf den Vergleich der Bewertungsmethoden und entwickeln keine strukturierte Vorgehensweise zur Auswahl einer Methode.

---

<sup>1</sup> Siehe <http://www.sei.cmu.edu/>

<sup>2</sup> Software Architecture Analysis Method (SAAM), Architecture Tradeoff Analysis Method (ATAM) und Active Reviews for Intermediate Design (ARID)

Im vorliegenden Beitrag wird auf Basis der skizzierten Arbeiten ein strukturierter Auswahlprozess für Bewertungsmethoden von Softwarearchitekturen entwickelt. Dazu werden zunächst die Beziehungen zwischen den existierenden Bewertungsmethoden untersucht und die Ergebnisse dieser Untersuchung in einer Taxonomie abgebildet. Anschließend werden die Kriterien identifiziert, die als sog. *Einstiegspunkte*, *Selektionskriterien* und *Filterkriterien* die Grundlage für die Entwicklung eines Auswahlprozesses bilden können. Ausgehend von den Einstiegspunkten wird dann exemplarisch ein strukturierter Auswahlprozess definiert.

## 2 Strukturierung der existierenden Methoden

### 2.1 Analyse der existierenden Bewertungsmethoden

Eine Literaturanalyse konnte achtzehn Methoden für die Bewertung von Softwarearchitekturen ermitteln. Um die Kriterien für eine Auswahl zu bestimmen, wurden zunächst die Beziehungen zwischen den unterschiedlichen Methoden analysiert, wobei zwei Beziehungstypen identifiziert werden konnten: Eine *Assoziation-Beziehung* beinhaltet, dass lediglich Teilaspekte einer Methode in andere Methode eingeflossen sind; durch die neue Kombination von Teilaspekten ist eine neue, eigenständige Methode entstanden. Die *Vererbung-Beziehung* verweist dagegen auf die Durchführung einer Spezialisierung; der Charakter der vererbenden Methode bleibt weitestgehend erhalten, da nur wenige Teilaspekte verändert bzw. erweitert wurden.

Zwischen sechs Methoden wurde die Existenz einer assoziativen Beziehung ermittelt. Abbildung 1 zeigt die sich ergebende Struktur; die Beschriftung der Verbindungen gibt die Teilaspekte an, die jeweils innerhalb der assoziierten Methode übernommen worden sind; die Pfeilspitze zeigt die Flussrichtung. In kursiver Schrift sind neben den Methoden jeweils die innerhalb der Methode *neu* entwickelten Konzepte und Modelle aufgeführt.

Die *Software Architecture Analysis Method (SAAM)* ist die erste am SEI entwickelte Methode; sie basiert auf der sog. Szenariotechnik, und bezieht insbesondere die Stakeholder<sup>3</sup> in die Bewertung mit ein (vgl. [KBA94, S. 81-90]). Die davon abgeleitete *Architecture Analysis Tradeoff Method (ATAM)*, die ebenfalls am SEI entwickelt worden ist, stellt eine Assoziation der SAAM dar, da auf Analyseverfahren und -techniken der SAAM zurückgegriffen wurde (vgl. [KKB97, S. 1]). Die ATAM selbst stellt dagegen eine eigenständige Methode dar, weil die Beziehungen zwischen den Entwurfsentscheidungen und den beeinflussten Qualitätsmerkmalen wesentlich detaillierter untersucht werden. Darüber hinaus wurden neue Konzepte wie z. B. der sog. Utility-Baum sowie die sog. Tradeoff Points und Sensitivity Points entwickelt (vgl. [CKK02, S. 36, 50-56]).

Die *Cost Benefit Analysis Method (CBAM)* ist eine Methode, die neben der Analyse nach Qualitätsmerkmalen auch eine Analyse der Softwarearchitektur hinsichtlich ihrer Wirtschaftlichkeit durchführt. Sie stellt eine Assoziation der ATAM dar, weil lediglich aus-

---

<sup>3</sup> Mit Stakeholder werden Personen oder Organisationen bezeichnet, die ein begründetes Interesse an der Softwarearchitektur besitzen. Dazu zählen u. a. die Benutzer des Systems, die Entwickler, die Administratoren sowie auch die Kunden (vgl. [BCK2003, S. 6]).

gewählte Aktivitäten der ATAM verwendet werden, um darauf aufgesetzt eine Analyse hinsichtlich der wirtschaftlichen Auswirkungen der Entwurfsentscheidungen durchzuführen (vgl. [BCK03, S. 313]). Die Szenariotechnik wurde dabei verfeinert, um die Auswirkungen und die Relevanz der Architekturentscheidungen festzulegen. Die Relevanz wird darüber hinaus verwendet, um auf Basis eines Kostenmodells den Return on Investment (ROI) zu bestimmen. Dieser reflektiert das Verhältnis von Nutzen zu den Kosten der Implementierung (vgl. [BCK03, S. 310-316]).

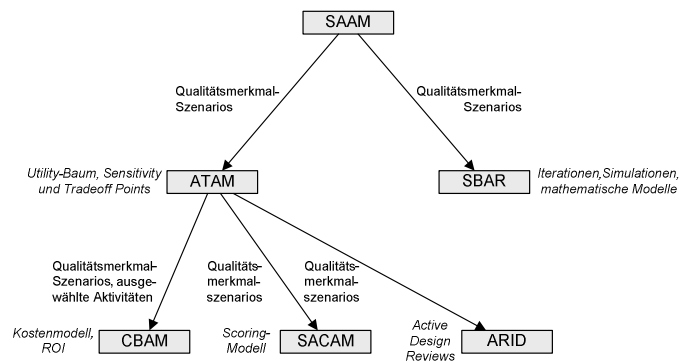


Abbildung 1: Assoziationen im Überblick

Die *Software Architecture Comparison Analysis Method (SACAM)* ist eine weitere SEI-Methode, die insbesondere für den Vergleich von mehreren Architekturalternativen verwendet werden kann. Sie stellt eine Assoziation zur ATAM dar, weil die Szenariotechnik die Grundlage für den Vergleich bildet. Den einzelnen Szenarios – bei SACAM Kriterien genannt – werden Werte zugewiesen, die den Erfüllungsgrad der Szenarios widerspiegeln. Diese Vorgehensweise wird bei SACAM Architektur-Scoring genannt (vgl. [SBV03, S. 21-22]).

Die *Active Reviews for Intermediate Design (ARID)* stellt eine Weiterentwicklung der ATAM dar. Die Szenariotechnik wird innerhalb der Methode mit den sog. Active Design Reviews (ADR) von *Parnas* und *Weiss* (vgl. [PW85]) kombiniert, weshalb *Clements* die ARID-Methode als ADR/ATAM-Hybrid bezeichnet (vgl. [CI00, S. 13]). Die Methode ist ausschließlich für die Bewertung von Teilentwürfen einer Architektur innerhalb der frühen Entwicklungsphase konzipiert.

Das *Scenario-Based Architecture Reengineering (SBAR)* schließlich basiert auf der Szenariotechnik und stellt eine Assoziation zur SAAM dar. Zum einen werden weitere Analysetechniken, wie z. B. mathematische Modelle und Simulationen verwendet, zum anderen werden bei der Szenario-Generierung die Stakeholder nicht berücksichtigt (vgl. [BZJ04, S. 312; BB98]).

Im Rahmen der Analyse von Vererbungsbeziehungen konnten sieben Spezialisierungen der SAAM und eine Spezialisierung der ATAM identifiziert werden (siehe Abbildung 2). Ein Beispiel für eine typische Spezialisierung ist die Fokussierung und Ausrichtung einer Methode auf ein bestimmtes Qualitätsmerkmal.

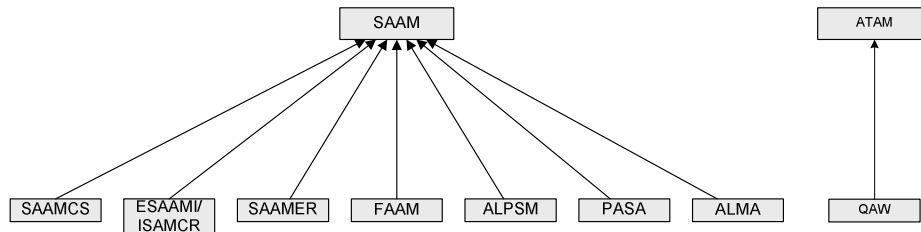


Abbildung 2: Vererbungen im Überblick

Zu den Spezialisierungen der SAAM zählen

- die *Software Architecture Analysis Method for Complex Scenarios (SAAMCS)* (siehe [LRV99]),
- die *Extending SAAM by Integration in the Domain (ESAAMI)/Integrating SAAM in Domain-centric and Reuse-based Development Processes (ISAAMCR)* [Mo99],
- die *Software Architecture Analysis Method for Evolution and Reusability (SAAMER)* (siehe [LBK97]),
- die *Family Architecture Assessment Method (FAAM)* (siehe [Do02]),
- die *Architecture Level Prediction of Software Maintenance (ALPSM)* (siehe [BB99]),
- die *Performance Assessment of Software Architecture (PASA)* (vgl. [Th05]) und
- die *Architecture-Level Modifiability Analysis (ALMA)*-Methode (siehe [BL04]).

Die *Quality Attribute Workshop (QAW)*-Methode (siehe [BE01]) ist die einzige Spezialisierung der ATAM.

Fünf der achtzehn untersuchten Methoden stellen eigenständige Entwicklungen dar und basieren auf keiner anderen Bewertungsmethode. Dabei handelt es sich um die Methoden *Software Architecture Evaluation (SAE)* (siehe [Th05]), *Software Architecture Analysis Method (SAAM)* (siehe [KBA94]), *Architecture Quality Assessment (AQA)* (siehe [HKL97]), *Attribute-Based Architectural Styles (ABAS)* (siehe [KK99]) und *Software Architecture Evaluation Model (SAEM)* (siehe [DOP99]).

## 2.2 Entwicklung einer Taxonomie zum Vergleich von Evaluationsmethoden

Im nächsten Schritt wurde eine Taxonomie zum Vergleich von Evaluationsmethoden entwickelt, indem die Vergleichskriterien von *Dobrica* und *Niemelä* auf der einen und von *Babar*, *Zhu* und *Jeffrey* auf der anderen Seite konsolidiert wurden.

Das Ziel der Klassifizierung bestand aber nicht nur im Vergleich der Methoden; vielmehr sollten auch entscheidungsunterstützende Eigenschaften der Methoden identifiziert werden. Die entwickelte Taxonomie umfasst zehn Kriterien, die in Tabelle 1 dargestellt und erläutert werden.

Kriterium	Beschreibung
Qualitätsmerkmal	Die Methoden betrachten eine unterschiedliche Anzahl von Qualitätsmerkmalen.
Berücksichtigung der Beziehungen zwischen den Qualitätsmerkmalen	Die meisten Methoden analysieren eine Softwarearchitektur nur für ein bestimmtes Qualitätsmerkmal zu einer Zeit. In der Praxis besteht jedoch häufig eine Wechselwirkung zwischen den Attributen einer Softwarearchitektur, z. B. gegensätzlich oder unterstützend, die nicht von allen Methoden berücksichtigt werden (vgl. [BZJ04, S. 312]).
Voraussetzungen für die Anwendung	Dieses Kriterium beschreibt die Voraussetzungen, die sich für die Anwendung einer bestimmten Methode ergeben. Von den organisatorischen und ressourcenabhängigen Anforderungen der Analysemethoden wird jedoch abstrahiert, da diese innerhalb des Entscheidungsprozesses nicht berücksichtigt werden.
Reifegrad und Validierung	Dieses Kriterium trifft eine Aussage über den Entwicklungsstand der Methode und in wie weit sie sich durch Projekte in der Praxis bewährt hat.
Detaillierungsgrad der Prozessbeschreibung	Das Kriterium beschreibt, inwieweit die Phasen und Aktivitäten des Evaluationsprozesses sowie die Richtlinien und Heuristik der Methode spezifiziert wurden. Eine gute Beschreibung dieser Elemente führt zu einer Verbesserung in der Anwendung der Methode (vgl. [BZJ04, S. 311]).
Ziel der Methode	Eine Evaluation von Softwarearchitekturen wird aus verschiedenen Gründen durchgeführt. Eine bestimmte Methode ist nicht für alle unterschiedlichen Anwendungsfälle geeignet, so dass dieses Kriterium eine hohe Relevanz für einen Auswahlprozess hat. Ein allgemeines Ziel lässt sich jedoch bei allen Methoden wieder finden: die Vorhersage der Qualität des Gesamtsystems auf Basis der Softwarearchitektur (vgl. [BZJ04, S. 312]). Im Speziellen lassen sich jedoch unterschiedliche Sichten und Ansätze identifizieren, um eine solche Aussage zu treffen. Diese Ansätze sind der Untersuchungsgegenstand dieses Kriteriums.
Projektphase	Innerhalb dieses Kriteriums wird zwischen einer frühen und späten Evaluation unterschieden, da es wichtig ist, die entsprechende Projektphase zu kennen, in der eine Methode <i>frühestens</i> angewendet werden kann (vgl. [BZJ04, S. 314]).
Bewertungsansatz	Das Kriterium beschreibt den grundlegenden Ansatz und die Techniken der Bewertung. Neben der Einteilung in Befragungs- und Messtechnik werden für jede Methode die speziellen Verfahren zur Bestimmung der Qualität identifiziert.
Stakeholder-Beteiligung	Das Konzept der Stakeholder stellt einen benutzerspezifischen Aspekt bei der Architekturanalyse dar. Für eine effektive Evaluation ist die aktive Beteiligung der Stakeholder essenziell (vgl. [CKK02, S. 64]). Zu jeder Methode wird daher die Aussage getroffen, ob eine Beteiligung der Stakeholder erfolgt oder nicht.
Erfahrung und Kenntnisse des Evaluationsteams	Die Evaluationsmethoden stellen unterschiedliche Anforderungen an die durchführenden Personen in Bezug auf deren Kenntnisse und Erfahrungen. Dieses Merkmal trifft, basierend auf den eingesetzten Bewertungstechniken, eine grobe Aussage, wie hoch die Anforderungen sind, um die Methode auszuführen.

Tabelle 1: Kriterien der Taxonomie

Auf Basis dieser umfassenden Taxonomie wurden die Bewertungsmethoden verglichen (siehe Tabellen 2 und 3 in [EHM07, S. 7-8]).

## 3 Entwicklung eines Auswahlprozesses

### 3.1 Identifizierung der Einstiegspunkte

Um den Auswahlprozess zu strukturieren, wurde eine Klassifikation der Kriterien entwickelt: Ein *Einstiegspunkt* ermöglicht direkt beim Einstieg die Eingrenzung der geeigneten Bewertungsmethoden, indem eine Selektion auf der Basis grundlegender Eigenschaften erfolgt. Von den zehn Kriterien der Taxonomie wurden die Kriterien „Qualitätsmerkmal“, „Projektphase“ und „Ziel der Bewertung“ als Einstiegskriterien identifiziert.

*Selektionskriterien* treffen im Gegensatz zu den Einstiegspunkten keine explizite Aussage, sondern enthalten nur implizite Informationen für den Selektionsprozess. Sie verfeinern somit die Klassifikation *nachgelagert* zu einem Einstiegspunkt (siehe Abbildung 3).



Abbildung 3: Grundanordnung des Auswahlprozesses

*Filterkriterien* können eingesetzt werden, um Mindestanforderungen für den Auswahlprozess zu definieren.

Um eine Auswahl für eine Methode zu treffen, müssen unter Umständen mehrere Kriterien in Kombination angewendet werden. Eine solche Kombination beschreibt somit einen bestimmten Auswahlprozess. Aufgrund des spezifischen Kontexts sind verschiedene Kombinationen denkbar, so dass durchaus mehrere Auswahlprozesse existieren können. Daher können nur allgemeine Aussagen bezogen auf die Kriterienarten und deren Anordnung getroffen werden.

Insgesamt ergibt sich das Vorgehen (vgl. auch Abbildung 3), dadurch dass beim Durchlauf der einzelnen Kriterien eine graduelle Verfeinerung der Betrachtung stattfindet und daher die Menge der in Frage kommenden Methoden reduziert wird. Die Einstiegspunkte stellen dabei den ersten Ansatzpunkt für die Verfeinerung der Betrachtung dar und bilden somit den ersten Schritt des Auswahlprozesses.

Je nach Einstiegspunkt und Kontext erfolgt zum Teil keine hinreichende Reduzierung der Auswahl. Im nächsten Schritt werden daher die Selektionskriterien genutzt, um eine weitere Einschränkung der Menge zu erreichen und sie im optimalen Fall auf eine Methode zu reduzieren. Im letzten Schritt werden die Filterkriterien dazu genutzt, die ausgewählte Methode bzw. die ausgewählten Methoden auf die Mindestanforderungen hin zu überprüfen.

Im Folgenden wird exemplarisch der Einstiegspunkt „Qualitätsmerkmal“ im Detail dargestellt.

### 3.2 Einstiegspunkt Qualitätsmerkmal

Der erste Auswahlzweig erfolgt anhand der Unterscheidung der unterstützten Qualitätsmerkmale auf Basis des ISO/IEC 9126-Qualitätsmodells. Das Modell wurde gewählt, um die Vergleichbarkeit der Qualitätsmerkmale zu gewährleisten (vgl. [II01, S. 1]). Ein weiterer Vorteil des Standards ist die Möglichkeit die Anforderungskriterien hinsichtlich ihrer Vollständigkeit zu validieren (vgl. [II01, S. 1]). Indem die Methoden den einzelnen Subcharakteristiken des Standards zugeordnet werden, ist transparent, welche Teile des ISO/IEC 9126-Qualitätsmodells durch die Methoden abgedeckt sind und in welchen Teilen keine spezielle Evaluationsmethode verfügbar ist (siehe Abbildung 4).

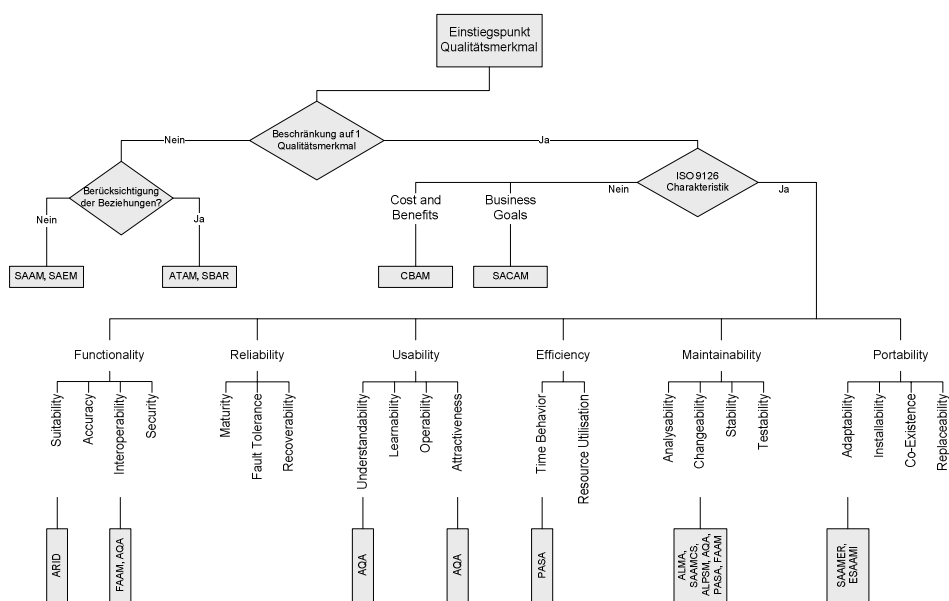


Abbildung 4: Einstiegspunkt Qualitätsmerkmal

Der erste Schritt im Auswahlprozess ist die Unterscheidung zwischen Methoden, die universell für mehrere Qualitätsmerkmale eingesetzt werden können, und Methoden, die sich auf ein bestimmtes Qualitätsmerkmal beschränken. Die Methoden, die nicht auf ein bestimmtes Merkmal beschränkt sind, sind SAAM, ATAM, SAEM und SBAR. Innerhalb dieser Auswahl erfolgt eine weitere Differenzierung einerseits in Methoden, die *gleichzeitig* mehrere Qualitätsmerkmale bei einer Evaluation untersuchen und die Beziehungen der Merkmale untereinander betrachten, und andererseits in Methoden, die jeweils nur ein Merkmal pro Evaluationsdurchlauf untersuchen. Festzustellen ist, dass ATAM und die SBAR die beiden einzigen Methoden sind, die die Beziehung der Merkmale untereinander berücksichtigen (vgl. [BZJ04, S. 312]). SAAM und SAEM betrachten dagegen jeweils nur ein Qualitätsmerkmal. Sowohl bei SBAR als auch bei ATAM müssen nicht zwangsläufig mehrere Qualitätsmerkmale bewertet werden. Streng genommen müssen daher neben SBAR und ATAM auch SAAM und SAEM gegenübergestellt werden.

Im nächsten Schritt des Auswahlprozesses erfolgt die Betrachtung der Methoden, die nur für ein bestimmtes Qualitätsmerkmal ausgelegt sind. Dazu werden die Methoden – sofern möglich – in das ISO/IEC 9126-Qualitätsmodell eingeordnet, um durch eine einheitliche Terminologie die Vergleichbarkeit zu gewährleisten und die Qualitätsmerkmale zu identifizieren, die nicht durch eine spezielle Evaluationsmethode abgedeckt sind.

Zunächst erfolgt eine Trennung in einerseits die Methoden, die in das ISO/IEC 9126-Qualitätsmodell eingeordnet werden können und andererseits die Methoden, die nicht eingeordnet werden können:

CBAM beurteilt eine Architektur im Hinblick auf Kosten-/Nutzen-Aspekte. Im ersten Schritt werden – ähnlich wie bei ATAM – die Auswirkungen der Entwurfsentscheidungen anhand der regulären Qualitätsmerkmale und auf Basis eines Kostenmodells die wirtschaftlichen Auswirkungen betrachtet (vgl. [BCK03, S. 310-314]).

SACAM vergleicht und bewertet mehrere Architekturalternativen anhand von Kriterien, die aus den Geschäftszielen abgeleitet sind (vgl. [SBV03, S. 21]).

Im letzten Schritt erfolgt die Betrachtung der Methoden, die in das ISO/IEC 9126-Qualitätsmodell eingeordnet werden können. Da die Methoden auf unterschiedlichen Qualitätsmodellen und Terminologien basieren, erfolgt eine Einordnung der innerhalb einer Methode referenzierten Qualitätsmerkmale in das einheitliche ISO/IEC 9126-Qualitätsmodell. Die Einordnung basiert auf der Semantik, die der Standard für jedes Qualitätsmerkmal vorgibt.

Bezüglich der sechs Hauptkriterien des ISO/IEC 9126-Qualitätsmodells fällt auf, dass der Bereich Zuverlässigkeit (Reliability) – im Hinblick auf eine Bewertung – von keiner Methode abgedeckt wird. Im Bereich Funktionalität (Functionality) werden zwei Kriterien durch eine Analyseverfahren berücksichtigt. ARID bewertet Teile eines Architekturentwurfs auf ihre Angemessenheit (Suitability) (vgl. [CKK02, S. 241]).

Das zweite Teilkriterium, welches durch eine Methode abgedeckt wird, ist die Interoperabilität (Interoperability). Dieses wird durch die FAAM abgedeckt (vgl. [Do02, S. 104]). Neben der FAAM wird Interoperabilität auch durch AQA überprüft. AQA verwendet hier jedoch den Begriff Offenheit (Openness) und meint damit die Offenheit gegenüber anderen Systemen (vgl. [HKL97, S. 4]). Bei dem Bereich Benutzbarkeit (Usability) wird lediglich ein Teilkriterium durch eine Methode abgedeckt. Der Teilaspekt Verständlichkeit (Understandability) ist direkt in AQA enthalten (vgl. [HKL97, S. 4]).

Die anderen drei Teilkriterien Erlernbarkeit (Learnability), Benutzung (Operability) und Attraktivität (Attractiveness) werden durch keine Methode erfasst.

Performanzaspekte werden durch die Charakteristik Effizienz (Efficiency) im ISO/IEC 9126-Qualitätsmodell beschrieben. PASA, eine Methode, die die Performanz einer Softwarearchitektur beurteilt, fokussiert im Speziellen den Teilbereich zeitliches Verhalten (Time Behavior), der bei PASA als Ansprechbarkeit (Responsiveness) bezeichnet wird

(vgl. [Th05, S. 64]). Der effiziente Einsatz der Ressourcen (Resource Utilisation) wird allerdings von keiner Methode berücksichtigt.

Im Teilbereich Änderbarkeit (Maintainability) sind mehrere Methoden zu finden. Sie beziehen sich alle auf den Aspekt Veränderbarkeit (Changeability). Die Teilkriterien Analysierbarkeit (Analysability), Stabilität (Stability) und Testbarkeit (Testability) werden von keiner Methode adressiert.

Fünf Methoden beschäftigen sich mit der Veränderbarkeit. Unterschieden wird zwischen der Modifizierbarkeit, d. h. der Weiterentwicklung des Systems, und der Wartung. Die Methoden ALMA (Modifiability), FAAM (Extensibility) und SAAMCS (Flexibility) beziehen sich auf die Weiterentwicklung des Systems (vgl. [BL04, S. 130; LRV99, S. 2; Do02, S. 109]). Im Fall von ALMA werden sogar die Aspekte Ausbesserung ausgeschlossen, die im Wartbarkeitsbegriff des ISO/IEC 9126-Standards definiert sind (vgl. [BL04, S. 130]).

Bei ALPSM wird ein sog. „Maintenance Profile“ erstellt, das Change-Szenarios beinhaltet (vgl. [BB99, S. 139]). Wie bereits zuvor erläutert, adressiert PASA Aspekte der Performanz. Einen Aspekt bildet die Skalierbarkeit: Die Skalierung eines Systems stellt für das ISO/IEC 9126-Qualitätsmodell eine spezielle Form der Veränderung dar, wie z. B. die Erhöhung der Benutzerzahl und des Durchsatzes.

Im Gegensatz dazu bewertet AQA sämtliche Aspekte der Wartbarkeit. Sie berücksichtigt sowohl eine Systemerweiterung als auch die Ausbesserung (vgl. [HKL97, S. 4]). Das letzte Hauptmerkmal des ISO/IEC 9126-Qualitätsmodells ist der Bereich der Übertragbarkeit (Portability). Ähnlich wie bei dem Bereich der Wartung existieren mehrere Methoden für einen bestimmten Teilbereich, während die anderen Subkriterien von keiner Methode berücksichtigt werden. Dies sind die Kriterien Installierbarkeit (Installability), Co-Existenz (Co-existence) und die Austauschbarkeit (Replaceability). Für das Kriterium Anpassbarkeit (Adaptability) sind die Methoden SAAMER und ESAAMI/ISAAMCR verfügbar. Sie beschäftigen sich mit dem Aspekt der Wiederverwendung, der keine Erweiterung der Funktionalität darstellt (vgl. [Mo99, S. 3; LBK97, S. 1]). Dies ist der Hauptunterschied zu dem Subkriterium Veränderbarkeit, das ebenfalls eine funktionale Erweiterung berücksichtigt (vgl. [II01, S. 10]).

Der Selektionsprozess anhand des Einstiegspunktes Qualitätsmerkmal führt zu einer Einteilung einerseits in Methoden, die universell in Bezug auf ein oder mehrere Qualitätsmerkmale eingesetzt werden können, und andererseits in Methoden, die speziell für ein bestimmtes Merkmal konzipiert sind. Es ist daher leicht erkennbar, ob eine spezielle Methode für den gewünschten Qualitätsbereich verfügbar ist, oder ob eine universelle Methode gewählt werden muss. Ein weiterer Mehrwert liegt in der Einteilung in das ISO/IEC 9126-Qualitätsmodell, da in Bezug auf eine vollständige Erfassung der Softwareproduktqualität unmittelbar abgelesen werden kann, für welche Aspekte der Qualität eine Methode entwickelt wurde und an welcher Stelle noch Bedarf nach einer Methode existiert.

## 4 Fazit

Die Softwarearchitektur eines Anwendungssystems beeinflusst entscheidend die Qualitätsmerkmale eines Anwendungssystems. Die Bewertung der gewählten Architektur bzw. der getroffenen Entwurfsentscheidungen erlaubt deshalb bereits vor der eigentlichen Implementierung eine Einschätzung der grundlegenden Qualitätsmerkmale des zu entwickelnden Systems. Da die Bewertung mit vergleichsweise geringen Kosten durchführbar ist, sollte sie fester Bestandteil der Konstruktion eines Softwaresystems werden.

Die relativ große Anzahl der entwickelten Bewertungsverfahren mit ihren individuellen Eigenschaften erschwert allerdings die Auswahl des oder der in einem konkreten Projekt einzusetzenden Verfahren. Im vorliegenden Beitrag wurde deshalb ein strukturierter und dabei leicht zu handhabender Auswahlprozess für Bewertungsverfahren exemplarisch am Beispiel eines Einstiegspunktes vorgestellt. Er basiert auf Entscheidungsbäumen, an deren Blättern jeweils auf eine bestimmte Methode oder eine Kombination von Methoden verwiesen wird. Die Bäume zeichnen sich dadurch aus, dass sie – bezogen auf ihren jeweiligen Einstiegspunkt – die universellen von den speziellen Methoden trennen. Aufgezeigt wurde dies im Beitrag für den Einstiegspunkt des Qualitätsmerkmals. Darüber hinaus wurde im Rahmen eines Forschungsberichts der Einstiegspunkt „Projektphase“ behandelt (siehe [EHM07]).

Das Ergebnis eines Durchlaufs durch den Auswahlprozess ist als eine Empfehlung einzuordnen. Zeigt der Baum des Einstiegspunktes gezielt auf eine Methode, lässt sich dennoch nicht ausschließen, dass diese Methode nicht für eine Anwendung geeignet ist, ohne dass die organisationspezifischen Aspekte der Softwarearchitektur beleuchtet werden. Sowohl in Bezug auf die Bewertungsmethoden als auch den Auswahlprozess sind daher weiterführende Arbeiten erforderlich.

## Literatur

- [BB98] Bengtsson, PerOlof; Bosch, Jan: Scenario-Based Software Architecture Reengineering. In: Proceedings of the 5th International Conference on Software Reuse. IEEE Computer Society Press 1998, S. 308-317.
- [BB99] Bengtsson, PerOlof; Bosch, Jan: Architecture Level Prediction of Software Maintenance. In: Proceedings of the 3rd European Conference on Maintenance and Reengineering, IEEE Computer Society Press 1999, S. 139-147.
- [BCK03] Bass, Len; Clements, Paul; Kazman, Rick: Software Architecture in Practice. 2. Aufl., Addison-Wesley, Boston 2003.
- [BE01] Barbacci, Mario; Ellison, Robert; Stafford, Judith, Weinstock, Charles, Wood, Williams: Quality Attribute Workshops. Forschungsbericht CMU/SEI-2001-TR-010 des Software Engineering Institute, Carnegie Mellon University, Pittsburgh 2001.
- [BL04] Bengtsson, PerOlof; Lassing, Nico; Bosch, Jan; van Vliet, Hans: Architecture-level Modifiability Analysis (ALMA). In: Journal of Systems and Software 69 (2004), S. 129-147.
- [BZJ04] Babar, Muhammad Ali; Zhu, Liming; Jeffrey, Ross: A Framework for Classifying and Comparing Software Architecture Evaluation Methods. In: Software Engineering Conference Proceedings. IEEE Computer Society Press 2004, S. 309-318.

- [CKK02] Clements, Paul; Kazman Rick; Klein, Mark: Evaluating Software Architecture – Methods and Case Studies. Addison-Wesley, Boston 2002.
- [CI00] Clements, Paul: Active Reviews for Intermediate Design. Forschungsnotiz CMU/SEI-2000-TN-010 des Software Engineering Institute, Carnegie Mellon University, Pittsburgh 2000.
- [DN02] Dobrica, Liliana; Niemelä, Eila: A Survey on Software Architecture Analysis Methods. In: IEEE Transactions on Software Engineering 28 (2002) 7, S. 638-653.
- [DOP99] Dueñas, Juan C.; de Oliveira, William L.; de la Puente, Juan Antonio: A Software Architecture Evaluation Model. In: Proceedings of the Second International ESPRIT ARES Workshop on Development and Evolution of Software Architectures for Product Families, Springer-Verlag 1999, S. 148-157.
- [Do02] Dolan, Thomas J.: Architecture Assessment of Information-Systems Families, a Practical Perspective. In: Library Technische Universiteit Eindhoven, Proefschrift 2002, S. 1-238.
- [EHM07] Eicker, Stefan; Hegmanns, Christian; Malich, Stefan: Auswahl von Bewertungsmethoden für Softwarerarchitekturen. Forschungsbericht Nr. 14 des Institutes für Informatik und Wirtschaftsinformatik, Universität Duisburg-Essen, Essen 2007.
- [HKL97] Hilliard II, Richard F.; Kurland, Michael J.; Litvintchouk, Steven D.: MITRE's Architecture Quality Assessment. In: Proceedings of Software Engineering & Economics Conference (1997), S. 1-9.
- [II01] International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC): ISO/IEC 9126-1 - Software engineering - Product quality - Part 1: Quality model. International Standard ISO/IEC 9126-1:2001(E).
- [KBA94] Kazman, Rick; Bass, Len; Abowd, Gregory; Webb, Mike: SAAM: A Method for Analyzing the Properties of Software Architectures. In: Proceedings of the 16th International Conference on Software Engineering (1994), Sorrento, S. 81-90.
- [KK99] Klein, Mark; Kazman, Rick: Attribute-Based Architectural Styles. Forschungsbericht CMU/SEI-1999-TR-010 des Software Engineering Institute, Carnegie Mellon University, Pittsburgh 1999.
- [KKB97] Kazman, Rick; Klein, Mark; Barbacci, Mario; Longstaff, Tom; Lipson, Howard; Carriere, Howard: The Architecture Tradeoff Analysis Method. Software Engineering Institute Carnegie Mellon University, Pittsburgh 1997.
- [LBK97] Lung, Chung-Horng; Bot, Sonia; Kalaichelvan, Kalai; Kazman, Rick: An Approach to Software Architecture Analysis for Evolution and Reusability. In: Proceedings Cascon (1997), S. 1-11.
- [LRV99] Lassing, Nico; Rijsenbrij, Daan; van Vliet, Hans: On Software Architecture Analysis of Flexibility. Complexity of Changes: Size isn't Everything. In: Proceedings of the 2nd Nordic Workshop on Software Architecture, Ronneby 1999, S. 1-6.
- [Mo99] Molter, Georg: Integrating SAAM in Domain-Centric and Reuse-based Development Processes. In: Proceedings of the 2nd Nordic Workshop on Software Architecture, Ronneby 1999, S. 1-10.
- [PW85] Parnas, David; Weiss, David: Active Design Reviews: Principles and Practice. In: Proceedings of the Eight International Conference on Software Engineering (1985), S. 132-136.
- [SBV03] Stoemer, Christoph; Bachmann, Felix; Verhoef, Chris: SACAM: The Software Architecture Comparison Analysis Method. Forschungsbericht CMU/SEI-2003-TR-06 des Software Engineering Institute, Carnegie Mellon University, Pittsburgh 2003.
- [SG96] Shaw, Mary; Garlan, David: Software Architecture – Perspectives on an Emerging Discipline. Prentice Hall, Upper Saddle River, 1996.
- [Th05] Thiel, Steffen: A Framework to Improve the Architecture Quality of Software-Intensive Systems. Dissertation, Universität Duisburg-Essen, Essen 2005.