



ICB

Institut für Informatik und
Wirtschaftsinformatik

Stefan Eicker
Christian Hegmanns
Stefan Malich



Auswahl von Bewertungsmethoden für Softwarearchitekturen

14
ICB-RESEARCH REPORT

ICB-Research Report No.14

March 2007

Die Forschungsberichte des Instituts für Informatik und Wirtschaftsinformatik dienen der Darstellung vorläufiger Ergebnisse, die i. d. R. noch für spätere Veröffentlichungen überarbeitet werden. Die Autoren sind deshalb für kritische Hinweise dankbar.

The ICB Research Reports comprise preliminary results which will usually be revised for subsequent publications. Critical comments would be appreciated by the authors.

Alle Rechte vorbehalten. Insbesondere die der Übersetzung, des Nachdruckes, des Vortrags, der Entnahme von Abbildungen und Tabellen – auch bei nur auszugsweiser Verwertung.

All rights reserved. No part of this report may be reproduced by any means, or translated.

Authors' Address:

Stefan Eicker
Christian Hegmanns
Stefan Malich

Institut für Informatik und
Wirtschaftsinformatik (ICB)
Universität Duisburg-Essen
Universitätsstr. 9
D-45141 Essen

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

ICB Research Reports

Edited by:

Prof. Dr. Heimo Adelsberger
Prof. Dr. Peter Chamoni
Prof. Dr. Frank Dorloff
Prof. Dr. Klaus Echtele
Prof. Dr. Stefan Eicker
Prof. Dr. Ulrich Frank
Prof. Dr. Michael Goedicke
Prof. Dr. Tobias Kollmann
Prof. Dr. Bruno Müller-Clostermann
Prof. Dr. Klaus Pohl
Prof. Dr. Erwin P. Rathgeb
Prof. Dr. Rainer Unland
Prof. Dr. Stephan Zelewski

Managing Assistant and Contact:

Jürgen Jung

Institut für Informatik und
Wirtschaftsinformatik (ICB)
Universität Duisburg-Essen
Universitätsstr. 9
45141 Essen
Germany

Email: icb@uni-duisburg-essen.de

ISSN 1860-2770

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 muss 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.

Inhaltsverzeichnis

1	EINLEITUNG	1
2	STRUKTURIERUNG DER EXISTIERENDEN METHODEN	3
2.1	ANALYSE DER EXISTIERENDEN BEWERTUNGSMETHODEN	3
2.2	ENTWICKLUNG EINER TAXONOMIE ZUM VERGLEICH VON EVALUATIONS-METHODEN	6
3	ENTWICKLUNG EINES AUSWAHLPROZESSES	9
3.1	IDENTIFIZIERUNG DER EINSTIEGSPUNKTE	9
3.2	EINSTIEGSPUNKT QUALITÄTSMERKMAL	10
3.3	EINSTIEGSPUNKT PROJEKTPHASE	13
4	FAZIT.....	15
	LITERATUR	16

Abbildungsverzeichnis

Abbildung 1: Assoziationen im Überblick	3
Abbildung 2: Vererbungen im Überblick.....	4
Abbildung 3: Grundanordnung des Auswahlprozesses.....	9
Abbildung 4: Einstiegspunkt Qualitätsmerkmal	11
Abbildung 5: Einstiegspunkt Projektphase	13

Tabellenverzeichnis

Tabelle 1: Kriterien der Taxonomie.....	6
Tabelle 2: Taxonomie (Teil 1).....	7
Tabelle 3: Taxonomie (Teil 2).....	8

Abkürzungsverzeichnis

ABAS	Attribute-Based Architectural Styles
ADR	Active Design Reviews
ALMA	Architecture-Level Modifiability Analysis
ALPSM	Architecture Level Prediction of Software Maintenance
AQA	Architecture Quality Assessment
ARID	Active Reviews for Intermediate Design
ATAM	Architecture Analysis Tradeoff Method
CBAM	Cost Benefit Analysis Method
ESAAMI	Extending SAAM by Integration in the Domain
FAAM	Family Architecture Assessment Method
GQM	Goal Question Metrics
ISAAMCR	Integrating SAAM in Domain-centric and Reuse-based Development Processes
PASA	Performance Assessment of Software Architecture
QAW	Quality Attribute Workshop
QM	Qualitätsmerkmal
ROI	Return on Investment
SAAM	Software Architecture Analysis Method
SAAMER	Software Architecture Analysis Method for Evolution and Reusability
SAAMCS	Software Architecture Analysis Method for Complex Scenarios
SACAM	Software Architecture Comparison Analysis Method
SAE	Software Architecture Evaluation
SAEM	Software Architecture Evaluation Model
SBAR	Scenario-Based Architecture Reengineering
SEI	Software Engineering Institute

1 Einleitung

Eine Softwarearchitektur beschreibt die wesentlichen Softwareelemente eines Anwendungssystems, die Eigenschaften der Elemente und die Beziehungen zwischen ihnen (vgl. [BaCK03, S. 21; ShGa96, S. 1]). Sie ist die Grundlage für die Aufwandschätzung und Projektplanung sowie auch 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. [CIKK02, 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. [CIKK02, 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. [CIKK02, S. 23-24, S. 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 [CIKK02, S. 255 ff.]). Sie beziehen aber nur am Software Engineering Institute (SEI)¹ entwickelte Bewertungsmethoden² ein. Außerdem wird die Auswahl einer Methode nicht durch einen strukturierten Prozess unterstützt.

Dobrica und *Niemelä* haben eine Klassifizierung entwickelt und insgesamt acht Bewertungsmethoden einander gegenübergestellt (siehe [DoNi02]). Die Klassifizierung gibt einige Richtlinien an, die den Vergleich der Methoden im Kontext der Bewertung einer Architektur unterstützen sollen (vgl. [DoNi02, 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. [DoNi02, 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 [BaZJ04]). Ziel des Vergleichs war eine ausführliche Beschreibung der Merkmale einer Methode sowie die Erfassung der Aspekte einer Methode, die im Allgemeinen bekannt, jedoch nur un-

¹ Siehe <http://www.sei.cmu.edu/>

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

Auswahl von Bewertungsmethoden für Softwarearchitekturen

zureichend oder informell beschrieben worden sind, so dass primär die Gemeinsamkeiten und Unterschiede der Methoden identifiziert werden können (vgl. [BaZ]04, S. 309-310]). Mit Ausnahme der szenario-spezifischen Kriterien sind alle Kriterien von *Dobrica* und *Niemelä* auch in diesem Vergleich zu finden. Darüber hinaus verwenden *Dobrica* und *Niemelä* zehn weitere Kriterien, wie z. B. das Ziel und die Prozessunterstützung der Methode (vgl. [BaZ]04, 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.

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 eine 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 der strukturierte 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.

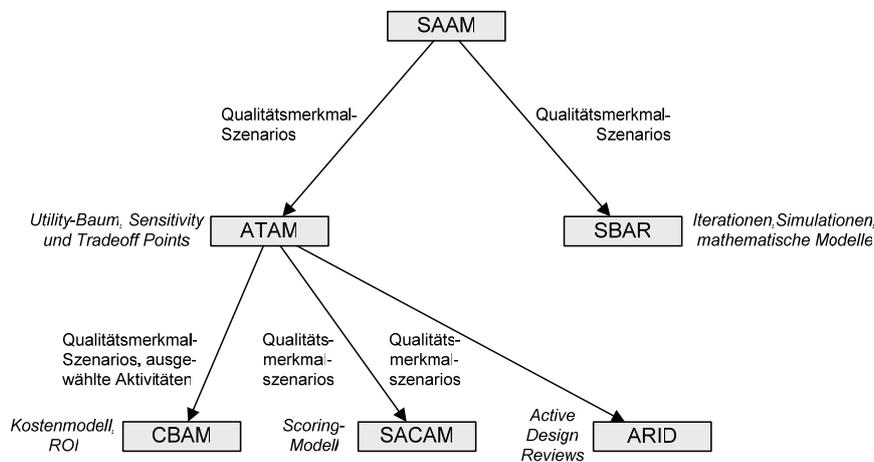


Abbildung 1: Assoziationen im Überblick

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 Stakeholder³ in die Bewertung mit ein (vgl. [KBAW94, 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. [KKBL97, S. 1]). Die ATAM selbst stellt dagegen eine eigenständige Methode dar, weil die Beziehungen zwischen den Entwurfsentscheidungen und den beeinflussten Qualitätsmerkmalen wesentlich detaillierter untersucht

³ 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. [BaCK2003, S. 6]).

Auswahl von Bewertungsmethoden für Softwarearchitekturen

werden. Darüber hinaus wurden neue Konzepte wie z. B. der sog. Utility-Baum sowie die sog. Tradeoff Points und Sensitivity Points entwickelt (vgl. [CIKK02, 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 ausgewählte Aktivitäten der ATAM verwendet werden, um darauf aufgesetzt eine Analyse hinsichtlich der wirtschaftlichen Auswirkungen der Entwurfsentscheidungen durchzuführen (vgl. [BaCK03, 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. [BaCK03, S. 310-316]).

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. [StBV03, 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. [PaWe85]) kombiniert, weshalb Clements die ARID-Methode als ADR/ATAM-Hybrid bezeichnet (vgl. [Clem00, S. 13]). Die Methode ist ausschließlich für die Bewertung von Teilentwürfen einer Architektur innerhalb der frühen Entwicklungsphase konzipiert.

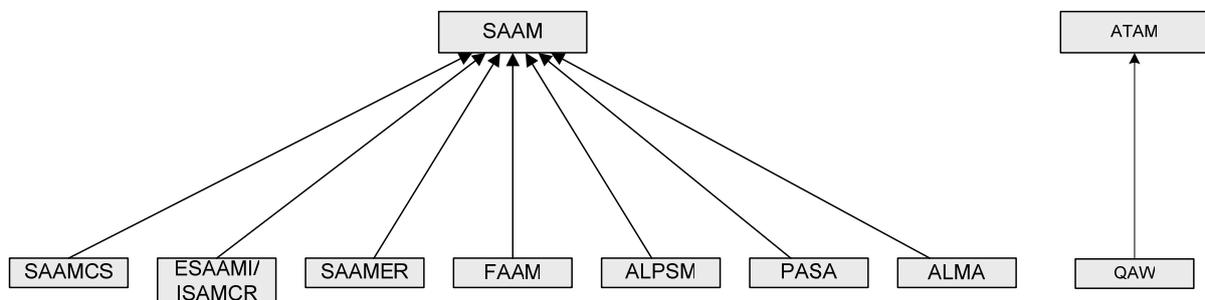


Abbildung 2: Vererbungen im Überblick

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. [BaZJ04, S. 312; BeBo98]).

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.

Zu den Spezialisierungen der SAAM zählen

- die *Software Architecture Analysis Method for Complex Scenarios (SAAMCS)* (siehe [LaRV99]),
- die *Extending SAAM by Integration in the Domain (ESAAMI) / Integrating SAAM in Domain-centric and Reuse-based Development Processes (ISAAMCR)* (siehe [Molt99]),
- die *Software Architecture Analysis Method for Evolution and Reusability (SAAMER)* (siehe [LBKK97]),
- die *Family Architecture Assessment Method (FAAM)* (siehe [Dola02]),
- die *Architecture Level Prediction of Software Maintenance (ALPSM)* (siehe [BeBo99]),
- die *Performance Assessment of Software Architecture (PASA)* (siehe [Thie05]) und
- die *Architecture-Level Modifiability Analysis (ALMA)* (siehe [BLBVO4]).

Die Quality Attribute Workshop (QAW)-Methode (siehe [BESW01]) 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 [Thie05]), *Software Architecture Analysis Method (SAAM)* (siehe [KBAW94]), *Architecture Quality Assessment (AQA)* (siehe [HiKL97]), *Attribute-Based Architectural Styles (ABAS)* (siehe [KIKa99]) und *Software Architecture Evaluation Model (SAEM)* (siehe [DuOP99]).

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 Strukturen der Methoden identifiziert werden. Die Spezialisierungen der SAAM wurden aufgrund der marginalen Unterschiede zur „Muttermethode“ nur im Auswahlprozess, nicht aber in der Taxonomie berücksichtigt.

Die entwickelte Taxonomie umfasst zehn Kriterien; diese werden in Tabelle 1 vorgestellt und erläutert.

Kriterium	Beschreibung
Qualitätsmerkmal	Die Methoden berücksichtigen eine unterschiedliche Anzahl von Qualitätsmerkmalen.
Berücksichtigung der Beziehungen zwischen den QM	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. Konfliktfeld oder unterstützend, die nicht von allen Methoden berücksichtigt werden (vgl. [BaZi04, 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 – sofern möglich – 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. [BaZi04, 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 Systemqualität auf Architekturebene (vgl. [BaZi04, 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 angewendet werden kann (vgl. [BaZi04, S. 314]).
Bewertungsansatz	Das Kriterium beschreibt den zugrunde liegenden Ansatz und die Techniken der Bewertung. Neben der Einteilung in Befragungs- und Messtechnik werden für jede Methode die speziellen Verfahren zur Bestimmung der Systemqualität identifiziert.
Stakeholder-Beteiligung	Das Konzept der Stakeholder stellt einen benutzerspezifischen Aspekt bei der Architekturanalyse dar. Für eine hochqualitative Evaluation ist die aktive Beteiligung der Stakeholder essenziell (vgl. [CIKK02, 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

Die beiden Tabellen zwei und drei zeigen die vollständige Taxonomie unterteilt in jeweils vier Methoden.

Kriterium	SAAM + Spez.	ATAM	ARID	SBAR
Qualitätsmerkmale	Modifizierbarkeit (+ Merkmale der Spezialisierungen)	Verschiedene	Entwurfstauglichkeit	Verschiedene
Berücksichtigung von Beziehungen	Nein	Ja	Nein	Ja
Voraussetzungen für die Anwendung	Keine besonderen Anforderungen, leicht anzuwendende Methode	Durch die Analyse- und Befragungsphase, hoher Ressourcenbedarf	Keine besonderen Anforderungen	Implementierte Architektur [BaZ]04, S. 316]
Reifegrad / Validierung	Ausgereift, erstmals 1996 beschrieben, Anwendung in vielen Bereichen	Sehr ausgereift, mittlerweile in 2. Version, Fallstudien und Trainings, validiert in vielen Projekten	Keine Angaben, verwendet jedoch validierte Techniken in Kombination (ADR + ATAM)	Keine Weiterentwicklung, keine Angaben zur Validierung
Detaillierungsgrad der Prozessbeschreibung	Ausführlich, inklusive Fallstudie, [CIKK02, S. 239]	Ausführlich, inklusive Fallstudie	Ausführlich, inklusive Fallstudie	Nur Konzept, keine detaillierte Beschreibung der einzelnen Schritte, keine Fallstudie
Ziel der Methode	Tauglichkeit (Suitability), Risiko	Sensitivity, Tradeoff, Risiko	Entwurfstauglichkeit	Ermittlung des Potenzials zur Erreichung des Qualitätsmerkmals
Projektphase	Früh	Früh	Früh, Entwurf der Komponenten	Spät, Systemerweiterung, Reengineering
Bewertungsansatz	Szenarios	Utility-Baum, Szenarios	Szenarios + ADR	Abhängig vom Attribut, Szenarios, math. Modelle, Simulation und auf Erfahrung basierte Deutung
Beteiligung Stakeholder	Ja	Ja	Ja	Nein, nur Designer
Erfahrung und Kenntnisse des Evaluationsteams	Leicht anzuwenden, kaum Erfahrung nötig	Hohe Anforderung an das Evaluationsteam wegen Aufstellen des Utility-Baums und der Identifizierung der Architekturansätze	Hohe Anforderungen an das Evaluationsteam, Erfahrung zur Bewertung der Ansätze nötig	Relativ hoch, durch Auswahl des richtigen Bewertungsansatzes

Tabelle 2: Taxonomie (Teil 1)

Auswahl von Bewertungsmethoden für Softwarearchitekturen

Kriterium	CBAM	SACAM	SAEM	AQA
Qualitätsmerkmale	Kosten und Nutzen	Abgeleitet von den Geschäftszielen	Externe und Interne Qualitätskriterien	Understandability, Feasibility, Openness, Maintainability, Evolvability und Client Satisfaction
Berücksichtigung von Beziehungen	Nein	Nein	Nein	Nein
Vorraussetzungen für die Anwendung	Teilweise Durchführung einer ATAM Analyse [BaCK03, S. 308-309]	Identifizierung der Geschäftsziele, Auswahl von Architekturkandidaten [StBVO3, S. 19]	Keine Angaben	Anforderungsanalyse [HiKL97, S. 5]
Reifegrad / Validierung	Ausgereift, umfangreiche Weiterentwicklung, [BaCK03, S.324]	Relativ neue Methode (2003), keine Angaben zur Validierung	Keine Angaben zum Reifegrad der Methode, keine Validierung	Prototyp, jedoch angewendet in Projekten der FAA und Air Force
Detaillierungsgrad der Prozessbeschreibung	Ausführlich, inklusive Fallstudie	Nur Phasen, keine Konzeptbeschreibung, viele Referenzen zu diversen Techniken	Keine Angaben	Nur Konzeptbeschreibung, keine Darstellung der 200 Maße [HiKL97, S. 6]
Ziel der Methode	Bestimmung der wirtschaftlichen Auswirkungen der Entwurfsentscheidungen, ROI [BaCK03, S. 313]	Vergleich der Tauglichkeit mehrerer Architekturalternativen [StBVO3, S. 1]	Vorhersage der Systemqualität [DoNi02, S. 647]	Objektivierte Analysemethode, Unterstützung für Entscheidungsträger [HiKL97, S. 1-2]
Projektphase	Früh	Früh	Spät, da metrische Methode	Spät, da metrische Methode
Bewertungsansatz	Wert des Szenarios (Utility), Kostenmodell, ROI [BaCK03, S. 310-314]	Kriterien, Szenarios, Direktiven, Scoringmodell [StBVO3, S. 21]	Metriken, GQM [DoNi02, S. 647]	Befragungstechnik, 200 Maße
Beteiligung Stakeholder	Ja	Ja, Comparison Stakeholder, Candidate Stakeholder [StBVO3, S. 18]	Nein	Nein
Erfahrung und Kenntnisse des Evaluations-teams	Vergleichsweise hoch durch Bestimmung der Utilities [BaCK03, S. 324]	Wirtschaftliches Verständnis erforderlich, da Ableitung QMs von den Geschäftszielen	Expertenwissen wird benötigt, um die QMs auf die internen Attribute abzubilden	Hohe Anforderung an Architekten, da interaktive Befragung in Bezug auf die Maße

Tabelle 3: Taxonomie (Teil 2)

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 Einstiegskriterium 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 mit einem Einstiegspunkt; ein Beispiel sind Methoden, die die Szenariotechnik verwenden und die Anforderungen der Stakeholder berücksichtigen – im Gegensatz zu quantitativen Methoden, die die Ergebnisse nur auf einem implementierten System messen.



Abbildung 3: Grundanordnung des Auswahlprozesses

Die relevanten Informationen sind vielschichtig; dies führt dazu, dass die Selektionskriterien rein qualitativ betrachtet werden können und nicht sämtliche Selektionskriterien verwendet werden müssen, um eine Auswahl zu treffen.

Filterkriterien können eingesetzt werden, um Mindestanforderungen für den Auswahlprozess zu definieren. Ein Beispiel ist das Kriterium „Reifegrad und Validierung“; ein möglicher Anwendungsfall wäre, dass nur ausgereifte und hinreichend validierte Methoden zu einem Auswahlprozess zugelassen werden.

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), dass beim Durchlauf durch die einzelnen Kriterien eine graduelle Verfeinerung der Betrachtung stattfindet und dadurch 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 der erste Schritt des Auswahlprozesses. Je nach Einstiegspunkt und Kontext erfolgt 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 werden exemplarisch die Einstiegspunkte „Qualitätsmerkmal“ und „Projektphase“ 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. [ISIE01, S. 1]). Eine weitere Funktion des Standards ist die Validierung von Anforderungskriterien auf Vollständigkeit (vgl. [ISIE01, 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).

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 eine Beziehung der Merkmale untereinander berücksichtigen (vgl. [BaZJ04, S. 312]). SAAM und SAEM betrachten dagegen jeweils nur ein Merkmal.

SAAM, ursprünglich für das Merkmal der Modifizierbarkeit konzipiert, ist eine szenario-basierte Methode, die keine besonderen Anforderungen hat und relativ leicht eingesetzt werden kann (vgl. [CIKK02, S. 212]). Die Methode trifft eine Aussage darüber, inwieweit die Architektur für das jeweilige Qualitätsmerkmal angemessen ist.

Das Ziel von SAEM besteht darin, die Qualität des Systems zu bewerten; die Methode deckt deshalb den gesamten Bereich der internen und externen Qualitätskriterien gemäß des ISO/IEC 9126-Standards ab und verwendet die im Rahmen des Standards entwickelten Metriken zur Evaluation. Dies führt dazu, dass – im Gegensatz zu SAAM – das System implementiert sein muss, um die Metriken einzusetzen und die Architektur bewerten zu können.

SBAR wird im Kontext einer Systemerweiterung angewendet und setzt daher ebenfalls ein implementiertes System voraus. Im Gegensatz zur ATAM wird SBAR nur von Designern angewendet. Bei ATAM dagegen kommt die Szenariotechnik zum Einsatz. Neben den Architekturansätzen werden auch in einer separaten Befragungsphase die Anforderungen der Stakeholder analysiert.

Sowohl bei SBAR als auch bei ATAM müssen nicht zwangsläufig mehrere Qualitätsmerkmale bewertet werden. Streng genommen müssen daher SBAR und ATAM auch SAAM und SAEM gegenübergestellt werden: SBAR und SAEM sehen keine Beteiligung der Stakeholder vor; SAAM und ATAM sind hingegen am SEI entwickelt worden und weisen daher viele Gemeinsamkeiten auf, vor allem beim Einsatz der Szenariotechnik (vgl. [CIKK02, S. 43]). Während SAAM eine leicht anzuwendende Methode ist, hat eine ATAM-Analyse einen relativ hohen Ressourcenbedarf. Der Hauptunterschied liegt jedoch in der Kombination aus Analyse und Befragung [CIKK02, S. 268]: Bevor die Stakeholder „befragt“ werden, erfolgt eine Analyse der Architekturansätze auf Basis von Anforderungsszenarios. Das Ziel dabei besteht in der Identifizierung von Sensitivity und Tradeoff Points sowie von Risks und Non-Risks (vgl. [CIKK02, S. 45]).

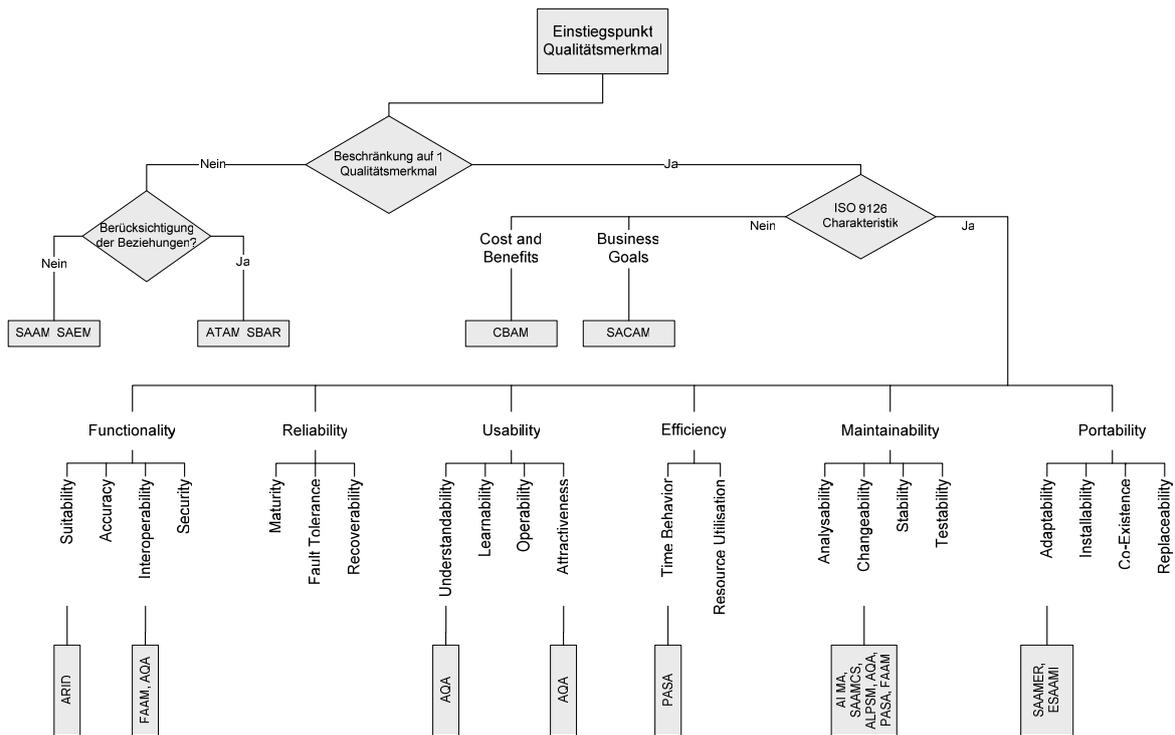


Abbildung 4: Einstiegsunkt Qualitätsmerkmal

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. [BaCK03, S. 310-314]).

SACAM vergleicht und bewertet mehrere Architekturalternativen anhand von Kriterien, die aus den Geschäftszielen abgeleitet sind (vgl. [StBV03, 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. [CIKK02, S. 241]).

Das zweite Teilkriterium, welches durch eine Methode abgedeckt wird ist Interoperabilität (Interoperability). Dieses wird durch die FAAM abgedeckt (vgl. [Dola02, S. 104]). Neben der

Auswahl von Bewertungsmethoden für Softwarearchitekturen

FAAM wird Interoperabilität auch durch AQA überprüft. AQA verwendet hier jedoch den Begriff Offenheit (Openness) und meint damit die Offenheit gegenüber anderen Systemen (vgl. [HiKL97, 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. [HiKL97, 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. [Thie05, 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 eigentlichen Wartung. Die Methoden ALMA (Modifiability), FAAM (Extensibility) und SAAMCS (Flexibility) beziehen sich auf die Weiterentwicklung des Systems (vgl. [BLBV04, S. 130; LaRV99, S. 2; Dola02, S. 109]). Im Fall von ALMA werden sogar die Aspekte Ausbesserung, die im Wartbarkeitsbegriff des ISO/IEC 9126-Standards definiert sind, ausgeschlossen (vgl. [BLBV04, S. 130]). Bei ALPSM wird ein sog. „Maintenance Profile“ erstellt, das Change-Szenarios beinhaltet (vgl. [BeBo99, 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-Qualitätsmodells eine spezielle Form der Veränderung dar, wie z. B. die Erhöhung der Nutzerzahl und des Durchsatzes.

Im Gegensatz dazu bewertet AQA sämtliche Aspekte der Wartbarkeit. Sie berücksichtigt sowohl eine Systemerweiterung als auch die Ausbesserung (vgl. [HiKL97, 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. [Molt99, S. 3; LBKK97, S. 1]). Dies ist der Hauptunterschied zu dem Subkriterium Veränderbarkeit, das ebenfalls eine funktionale Erweiterung berücksichtigt (vgl. [ISIE01, 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.

3.3 Einstiegspunkt Projektphase

Der Einstiegspunkt Projektphase bezieht sich auf die Phase, in der eine Methode angewendet wird. Unterschieden wird zunächst bezüglich des Evaluationszeitpunkts: Grundsätzlich ist eine Evaluation der Architektur zu jedem Zeitpunkt im Lebenszyklus eines Systems möglich. Es ist jedoch zwischen zwei generellen Varianten zu unterscheiden, der *frühen Evaluation* und der *späten Evaluation* (vgl. [CIKK02, S. 24]). Von einer frühen Evaluation wird gesprochen, wenn die Architektur eines Systems noch nicht vollständig implementiert oder spezifiziert ist und sich somit noch im Entwicklungsstadium befindet. Das primäre Ziel einer frühen Evaluation ist es festzustellen, ob potenzielle Risiken in Bezug auf die Erfüllung der Qualitätsanforderungen bestehen, die auf die frühen Entwurfsentscheidungen zurückzuführen sind. Je früher der Zeitpunkt innerhalb des Lebenszyklus liegt, umso kleiner ist die Gruppe der an der Evaluation Beteiligten.

Eine Methode, die eine frühe Evaluation unterstützt, sind die Active Reviews for Intermediate Design (ARID).

Die Evaluation einer Architektur wird als „spät“ bezeichnet, wenn der Entwurf der Architektur feststeht und die Architektur darüber hinaus implementiert ist. Eine späte Evaluation ist daher für zwei unterschiedliche Arten von Systemen relevant, zum einen Neuentwicklungen und zum anderen bereits bestehende Systemen, z. B. Legacy-Systemen oder auch Standardsoftwareprodukten (vgl. [CIKK02, S. 25]). Die Technik der Evaluation ist in beiden Fällen identisch; lediglich die zu überprüfenden Qualitätskriterien sind in der Regel unterschiedlich (vgl. [CIKK02, S. 25]).

Quantitative Messungen können nur durchgeführt werden, wenn eine Architektur implementiert ist oder wenn zumindest ein Prototyp vorliegt. Vor der Implementierung können dagegen nur qualitative Aussagen getroffen bzw. qualitative Schätzungen durchgeführt werden.

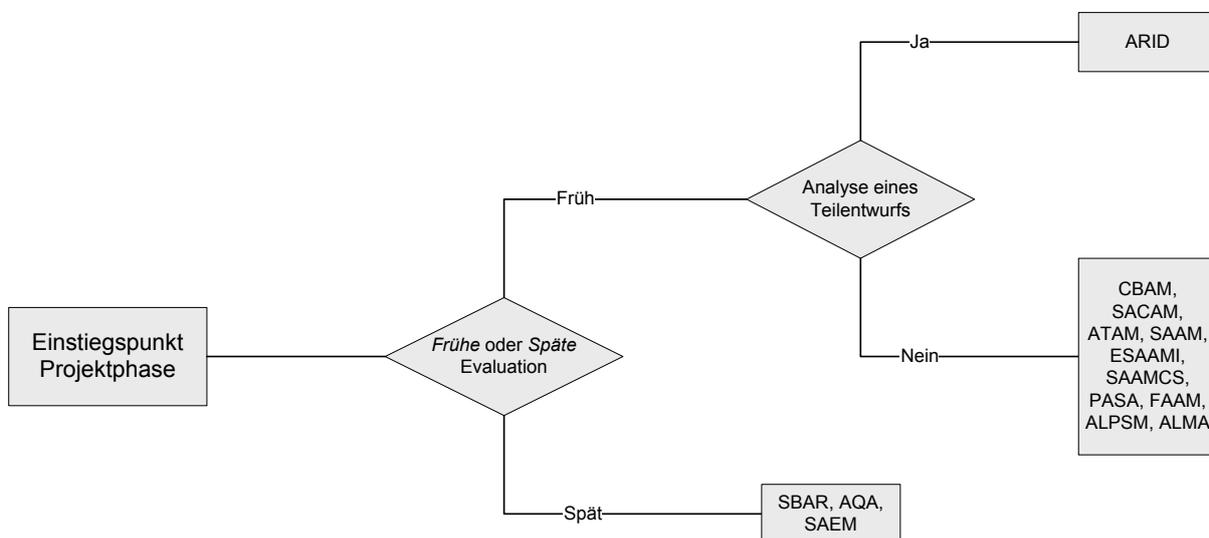


Abbildung 5: Einstiegspunkt Projektphase

Abbildung 5 zeigt den sich ergebenden Selektionsprozess dieses Einstiegspunktes im Überblick: Bei den Methoden des frühen Evaluationszeitpunktes erfolgt eine weitere Einteilung; auf der einen Seite werden die Methoden betrachtet, bei denen eine Analyse einer vollständig spezifizierten Architektur erfolgen kann, auf der anderen Seite die Methoden, die nur Teilentwürfe betrachten können und somit in einer sehr frühen Entwicklungsphase angewendet werden.

Auswahl von Bewertungsmethoden für Softwarearchitekturen

Die meisten der untersuchten Methoden sind für eine frühe Evaluation geeignet; unter diesen Methoden findet sich lediglich eine Methode, die *nicht* dafür ausgelegt ist, eine vollständig spezifizierte Architektur zu analysieren (siehe Abbildung 5).

Die übrigen Methoden können dafür genutzt werden, um eine vollständige Softwarearchitektur zu bewerten. Bei diesen Methoden handelt es sich um CBAM, SACAM, ATAM, SAAM und sämtliche hier angeführten Spezialisierungen von SAAM (SAAMER, ESAAMI/ISAAMCR, SAAMCS, PASA, FAAM, ALPSM, ALMA). Dies schließt jedoch nicht aus, dass diese Methoden nicht eingesetzt werden können, bevor die Architektur vollständig spezifiziert ist. Als Beispiel kann ATAM angeführt werden, da die Methode zwar grundsätzlich auf die Analyse vollständig spezifizierter Architekturen ausgerichtet ist, jedoch auch eingesetzt werden kann, um Entwurfsansätze zu bewerten (vgl. [CIKK02, S. 43-45]).

Die Methoden, die für eine späte Evaluation eingesetzt werden, sind SBAR, SAEM und AQA. SAEM und AQA basieren auf Metriken und Maßen. Die Ausprägungen dieser Maße können nur am implementierten System gemessen und nachvollzogen werden, so dass diese Methoden nur in einer „späten“ Phase angewendet werden können.

SBAR wird im Rahmen des Reengineering angewendet, was die Einordnung in eine späte Evaluation unterstreicht (vgl. [BeBo98, S. 1, 9]).

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 zudem 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 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 die Einstiegspunkte Qualitätsmerkmal und Projektphase.

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

- [BaCK03] Bass, Len; Clements, Paul; Kazman, Rick: Software Architecture in Practice. 2. Aufl., Addison-Wesley, Boston 2003.
- [BaZJ04] 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.
- [BeBo98] Bengtsson, PerOlof; Bosch, Jan: Scenario-Based Software Architecture Reengineering. In: Proceedings of the Fifth International Conference on Software Reuse. IEEE Computer Society Press 1998, S. 308-317.
- [BeBo99] 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.
- [BESW01] 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.
- [BLBV04] 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.
- [Clem00] Clements, Paul: Active Reviews for Intermediate Design. Forschungsnotiz CMU/SEI-2000-TN-010 des Software Engineering Institute, Carnegie Mellon University, Pittsburgh 2000.
- [ClKK02] Clements, Paul; Kazman Rick; Klein, Mark: Evaluating Software Architecture – Methods and Case Studies. Addison-Wesley, Boston 2002.
- [Dola02] Dolan, Thomas J.: Architecture Assessment of Information-Systems Families, a Practical Perspective. In: Library Technische Universiteit Eindhoven, Proefschrift 2002, S. 1-238.
- [DoNi02] Dobrica, Liliana; Niemelä, Eila: A Survey on Software Architecture Analysis Methods. In: IEEE Transactions on Software Engineering 28 (2002) 7, S. 638-653.
- [DuOP99] 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.
- [HiKL97] 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.
- [ISIE01] 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).
- [KBAW94] Kazman, Rick; Bass, Len; Abowd, Gregory; Webb, Mike: SAAM: A Methode for Analyzing the Properties of Software Architectures. In: Proceedings of the 16th International Conference on Software Engineering (1994), Sorrento, S. 81-90.
- [KKBL97] 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.

- [KlKa99] Klein, Mark; Kazman, Rick: Attribute-Based Architectural Styles. Forschungsbericht CMU/SEI-1999-TR-010 des Software Engineering Institute, Carnegie Mellon University, Pittsburgh 1999.
- [LaRV99] 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.
- [LBKK97] 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.
- [Molt99] 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.
- [PaWe85] Parnas, David; Weiss, David: Active Design Reviews: Principles and Practice. In: Proceedings of the Eight International Conference on Software Engineering (1985), S. 132-136.
- [ShGa96] Shaw, Mary; Garlan, David: Software Architecture – Perspectives on an Emerging Discipline. Prentice Hall, Upper Saddle River, 1996.
- [StBV03] 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.
- [Thie05] Thiel, Steffen: A Framework to Improve the Architecture Quality of Software-Intensive Systems. Dissertation, Universität Duisburg-Essen, Essen 2005.

Previously published ICB - Research Reports

2007

No 13 (February 2007)

Eicker, Stefan; Spies, Thorsten; Kahl, Christian: "Softwarevisualisierung im Kontext serviceorientierter Architekturen"

No 12 (February 2007)

Brenner, Freimut: "Cumulative Measures of Absorbing Joint Markov Chains and an Application to Markovian Process Algebras"

No 11 (February 2007)

Kirchner, Lutz: "Entwurf einer Modellierungssprache zur Unterstützung der Aufgaben des IT-Managements – Grundlagen, Anforderungen und Metamodell"

No 10 (February 2007)

Schauer, Carola; Strecker, Stefan: "Vergleichende Literaturstudie aktueller einführender Lehrbücher der Wirtschaftsinformatik: Bezugsrahmen und Auswertung"

No 9 (February 2007)

Strecker, Stefan; Kuckertz, Andreas; Pawlowski, Jan M.: "Überlegungen zur Qualifizierung des wissenschaftlichen Nachwuchses: Ein Diskussionsbeitrag zur (kumulativen) Habilitation"

No 8 (February 2007)

Frank, Ulrich; Strecker, Stefan; Koch, Stefan: "Open Model – Ein Vorschlag für ein Forschungsprogramm der Wirtschaftsinformatik (Langfassung)"

2006

No 7 (December 2006)

Frank, Ulrich: "Towards a Pluralistic Conception of Research Methods in Information Systems Research"

No 6 (April 2006)

Frank, Ulrich: "Evaluation von Forschung und Lehre an Universitäten – Ein Diskussionsbeitrag"

No 5 (April 2006)

Jung, Jürgen: "Supply Chains in the Context of Resource Modelling"

No 4 (February 2006)

Lange, Carola: "Development and status of the Information Systems / Wirtschaftsinformatik discipline: An interpretive evaluation of interviews with renowned researchers, Part III – Results Wirtschaftsinformatik Discipline"

2005

No 3 (December 2005)

Lange, Carola: "Development and status of the Information Systems / Wirtschaftsinformatik discipline: An interpretive evaluation of interviews with renowned researchers, Part II – Results Information Systems Discipline"

No 2 (December 2005)

Lange, Carola: "Development and status of the Information Systems / Wirtschaftsinformatik discipline: An interpretive evaluation of interviews with renowned researchers, Part I – Research Objectives and Method"

No 1 (August 2005)

Lange, Carola: „Ein Bezugsrahmen zur Beschreibung von Forschungsgegenständen und -methoden in Wirtschaftsinformatik und Information Systems“

The Institute for Computer Science and Business Information Systems (ICB), located at the Essen Campus, is dedicated to research and teaching in Applied Computer Science, Information Systems as well as Information Management. The ICB research groups cover a wide range of expertise:

Research Group	Core Research Topics
Prof. Dr. H. H. Adelsberger Information Systems for Production and Operations Management	E-Learning, Knowledge Management, Skill-Management, Simulation, Artificial Intelligence
Prof. Dr. P. Chamoni MIS and Management Science / Operations Research	Information Systems and Operations Research, Business Intelligence, Data Warehousing
Prof. Dr. F.-D. Dorloff Procurement, Logistics and Information Management	E-Business, E-Procurement, E-Government
Prof. Dr. K. Echtele Dependability of Computing Systems	Dependability of Computing Systems
Prof. Dr. S. Eicker Information Systems and Software Engineering	Process Models, Software-Architectures
Prof. Dr. U. Frank Information Systems and Enterprise Modelling	Enterprise Modelling, Enterprise Application Integration, IT Management, Knowledge Management
Prof. Dr. M. Goedicke Specification of Software Systems	Distributed Systems, Software Components, CSCW
Prof. Dr. T. Kollmann E-Business and E-Entrepreneurship	E-Business and Information Management, E-Entrepreneurship/ E-Venture, Virtual Marketplaces and Mobile Commerce, Online-Marketing
Prof. Dr. B. Müller-Clostermann Systems Modelling	Performance Evaluation, Modelling and Simulation, SAP Capacity Planning for R/3 and mySAP.com, Tools for Queueing Network Analysis and Capacity Planning, Communication Protocols and Distributed Systems, Mobile Systems
Prof. Dr. K. Pohl Software Systems Engineering	Requirements Engineering, Software Quality Assurance, Software-Architectures, Evaluation of COTS/Open Source-Components
Prof. Dr.-Ing. E. Rathgeb Computer Networking Technology	Computer Networking Technology
Prof. Dr. R. Unland Data Management Systems and Knowledge Representation	Data Management, Artificial Intelligence, Software Engineering, Internet Based Teaching
Prof. Dr. S. Zelewski Institute of Production and Industrial Information Management	Industrial Business Processes, Innovation Management, Information Management, Economic Analyses