



Thema:

**Vergleich von Qualitätsbewertungsmethoden
für IT-Architekturen**

Studienarbeit

Arbeitsgruppe Wirtschaftsinformatik

Themensteller: Prof. Dr. Hans-Knud Arndt
Betreuer: Dipl.-Wirt.-Inform. Bastian Grabski

vorgelegt von: Frank Eichler

Abgabetermin: 04. August 2009

Inhaltsverzeichnis

VERZEICHNIS DER ABKÜRZUNGEN UND AKRONYME	III
ABBILDUNGSVERZEICHNIS	IV
TABELLENVERZEICHNIS	V
1 ARCHITEKTURANALYSE ALS WEG ZUR ERFOLGREICHEN ENTWICKLUNG	1
2 DER ARCHITEKTURBEGRIFF IN DER INFORMATIK	3
2.1 Begriffsbestimmung	3
2.2 Bedeutung von Architekturen bei IT-Projekten	6
2.3 Qualitätsbewertung von Architekturen	7
2.3.1 <i>Notwendigkeit zur Bewertung der Qualität</i>	7
2.3.2 <i>Zeitpunkt der Bewertung</i>	7
2.3.3 <i>Beteiligte Personen</i>	8
2.3.4 <i>Qualitätsmerkmale</i>	8
3 QUALITÄTSBEWERTUNGSMETHODEN	11
3.1 Allgemeine Informationen	11
3.2 Übersicht der Qualitätsbewertungsmethoden	11
3.2.1 SAAM	11
3.2.2 ATAM	13
3.2.3 ARID	16
3.2.4 QAW	17
3.2.5 SAE	17
3.2.6 AQA	18
3.2.7 ALMA	19
3.2.8 APSM	19
3.2.9 PASA	20
3.2.10 FAAM	21
3.2.11 CBAM	22
3.2.12 SAEM	23
3.2.13 SACAM	24
3.2.14 <i>Weitere Qualitätsbewertungsmethoden</i>	25
3.3 Ursprung der Methoden	26
4 ENTWICKLUNG DES KRITERIENKATALOGS FÜR DEN VERGLEICH	28
5 VERGLEICH DER QUALITÄTSBEWERTUNGSMETHODEN	31
6 ERGEBNISSE	44
LITERATURVERZEICHNIS	46

Verzeichnis der Abkürzungen und Akronyme

ALMA	Architecture Level Modifiability Analysis
APSM/ALPSM	Architecture (Level) Prediction of Software Maintenance
AQA	Architecture Quality Assessment
ARID	Active Reviews for Intermediate Designs
AS	Architecture Strategies (Architekturstrategien)
ASAAM	Aspectual Software Architecture Analysis Method
ATAM	Architecture Trade-Off Analysis Method
CBAM	Cost-Benefit Analysis Method
ESAAMI	Extending SAAM by Integration in the Domain
FAAM	Family Architecture Assessment Method
GQM	Goal Question Metric
ISAAMCR	Integrating SAAM in Domain-centric and Reuse-based Development Processes
IT	Informationstechnologie
PASA	Performance Assessment of Software Architectures
QA	Quality Attribute (Qualitätsattribut)
QAW	Quality Attribute Workshop
ROI	Return on Investment
SAAM	Software Architecture Analysis Method
SAAMCS	Software Architecture Analysis Method Founded on Complex Scenarios
SAAMER	Software Architecture Analysis Method for Evolution and Reusability
SACAM	Software Architecture Comparison Analysis Method
SAE	Software Architecture Evaluation
SAEM	Software Architecture Evaluation Model
SBAR	Scenario-based Architecture Reengineering
SEI	Software Engineering Institute

Abbildungsverzeichnis

Abb. 2.1: Interaktionen bei einer Client/Server-Architektur	5
Abb. 3.1: Beispiel für einen Nutzenbaum (“utility tree“).....	14
Abb. 3.2: Beispiel für die Analyse eines Architekturansatzes mit einem Szenario.....	15
Abb. 3.3: Ablauf der ARID Methode.....	16
Abb. 3.4: CBAM Ablauf.....	23
Abb. 3.5: SACAM Prozessübersicht	24
Abb. 3.6: Spezialisierungen der SAAM	26

Tabellenverzeichnis

Tabelle 5.1: Vergleich der Qualitätsbewertungsmethoden (Teil 1).....	32
Tabelle 5.2: Vergleich der Qualitätsbewertungsmethoden (Teil 2).....	33
Tabelle 5.3: Vergleich der Qualitätsbewertungsmethoden (Teil 3).....	34
Tabelle 5.4: Vergleich der Qualitätsbewertungsmethoden (Teil 4).....	35
Tabelle 5.5: Vergleich der Qualitätsbewertungsmethoden (Teil 5).....	36
Tabelle 5.6: Vergleich der Qualitätsbewertungsmethoden (Teil 6).....	37

1 Architekturanalyse als Weg zur erfolgreichen Entwicklung

Die Entwicklung komplexer informationstechnischer Systeme ist im Laufe der Zeit immer schwieriger geworden und zumindest wenn man sich die Erfolgsaussichten für die Durchführung eines solchen Projektes ansieht, ähnelt ein solches Projekt einem Glücksspiel. Die Quote der Projekte, die als Misserfolge zu bezeichnen sind oder zumindest von großen Problemen gebremst werden, ist mit 72 Prozent zu bemessen (vgl. Herden et al. (2006), S. 1), wobei die Gründe dafür von schlechter Planung, über mangelnde Kommunikation und aus dem Ruder laufende Budgets sehr vielfältig ausfallen können. Hinzu kommt natürlich die immer komplexere Natur solcher Systeme, die sich von monolithischen Lösungen, hin zu verteilten Systemen entwickelt haben und immer höhere Anforderungen an die Systementwickler stellen. Doch es gibt Wege, um die Erfolgchancen von IT-Projekten zu erhöhen. Dazu gehört die frühzeitige Analyse und Bewertung der, dem System zugrundeliegenden, Architektur.

“To put it bluntly, an architecture is a bet, a wager on the success of a system.”
(Clements et al. (2002), S. 19)

Diese Aussage beschreibt die Rolle der Architektur im Entwicklungsprozess sehr gut, die Architektur kommt einer Wette gleich, einem Einsatz auf den Erfolg des Systems. Die Architektur beeinflusst die späteren Eigenschaften des gesamten Systems, sie ist Grundlage für eine geordnete Entwicklung. Laut Clements et al. (2002, S. 2) kann sie als Vehikel für die Kommunikation unter den Stakeholdern, also den Interessenvertretern/Anspruchsgruppen die in unterschiedlicher Weise Anteil an der Architektur haben, dienen. Die Architektur ist eine Manifestation von frühen Designentscheidungen und stellt eine wiederverwendbare, übertragbare Abstraktion eines Systems dar. Die Qualität der Architektur ist also ein nützlicher Indikator für die spätere Systemqualität und somit sollte die Evaluation der Architekturqualität durchaus erstrebenswert sein.

In dieser Studienarbeit, werden Qualitätsbewertungsmethoden für Architekturen miteinander verglichen. Viele der vorgestellten Methoden stammen aus dem Gebiet der Softwarearchitekturbewertung, doch sie befassen sich nicht nur mit der Software. Die Hardware, die nötig ist um das System umzusetzen, sowie die notwendigen Infrastrukturen (Netzwerke o.ä.) werden ebenfalls in die Evaluation einbezogen, so dass der Begriff der IT-Architektur wohl eher passend ist.

Das wissenschaftliche Ziel der Arbeit, der Vergleich der vorhandenen Methoden zur Qualitätsbewertung von Architekturen, soll über den folgenden Aufbau der Arbeit verwirklicht werden:

- Erläuterung der Grundlagen,
- Vorstellung der zu vergleichenden Qualitätsbewertungsmethoden,
- Entwicklung eines Kriterienkatalogs für den Vergleich,
- Durchführung des Vergleichs und
- Zusammenfassung der Ergebnisse.

In dieser Arbeit werden Methoden vorgestellt und verglichen, die auf verschiedenen Bewertungsansätzen beruhen. Dazu gehören Szenario-basierte Methoden, Methoden die Befragungstechniken nutzen, Simulations- oder Modell-basierte Methoden und Methoden die Metriken und Messungen nutzen. Diese unterschiedlichen Ansätze bewirken weitere Unterschiede zwischen den Methoden, die über verschiedene Kriterien untersucht und in einem umfangreichen Vergleich hervorgehoben werden.

Zunächst folgen die Begriffsbestimmung und die Betrachtung der Grundlagen, auf denen die Arbeit basiert.

2 Der Architekturbegriff in der Informatik

2.1 Begriffsbestimmung

Der Begriff „Architektur“ taucht in der Informatik sehr häufig auf und wird auf unterschiedlichste Art und Weise in der Literatur genutzt. In allen Bereichen der Informationstechnologie (IT) wird diesem Begriff hohe Bedeutung zugewiesen, sei es die Architektur eines Softwarebausteins, der Aufbau und das Zusammenspiel von Netzwerken oder die Geschäftsarchitektur eines Unternehmens. Ursprünglich aus dem Griechischen stammend, setzt sich der Begriff aus ‚*arché*‘ (Anfang, Ursprung) und ‚*techné*‘ (Handwerk, Kunst, Fertigkeit) zusammen womit man ihm die Bedeutung „erste Kunst“ oder „oberste Fertigkeit“ zuordnen kann (vgl. Herden et al. (2006), S. 1). Die Architektur bildet also den Anfang allen Handwerks auch in der Informatik, sie dient zur Strukturierung von Systemen (Mengen von Elementen und Beziehungen zwischen diesen (vgl. Heinrich et al. (2004), S. 642)) und ermöglicht eine zielorientierte Entwicklung. So wird die Architektur auch im Lexikon der Wirtschaftsinformatik definiert:

„Die Architektur eines Objektsystems ist also ein abstraktes Modell dieses Objektsystems, das seine Funktionen und Schnittstellen beschreibt.“ (Heinrich et al. (2004), S. 72)

Im Wesentlichen soll es in dieser Arbeit um die Bewertung von IT-Architekturen (sie beschreiben die Gesamtheit der Informationsinfrastruktur eines Unternehmens) und genauer gesagt von Softwarearchitekturen (als bedeutende Komponente von IT-Architekturen) gehen (vgl. Heinrich et al. (2004), S. 346). Die Hardwarearchitektur (welche die Implementierung/Technologische Realisierung in Hardwaresystemen darstellt) spielt bei den meisten Methoden eine untergeordnete Rolle. Es folgen einige beispielhafte Definitionen der Softwarearchitektur:

„[Software Architektur:] Die grundlegende Organisation eines Systems, dargestellt durch dessen Komponenten, deren Beziehungen zueinander und zur Umgebung sowie den Prinzipien, die den Entwurf und die Evolution des Systems bestimmen.“ (Hasselbring (2006), S. 48)

„Softwarearchitekturen zeigen die Zusammenhänge zwischen Elementen von Softwaresystemen.“ (Herden et al. (2006), S. 2)

„Software-Architektur ist die Identifikation, Spezifikation und Dokumentation sowohl der statischen Struktur als auch der dynamischen Interaktion eines Software-Systems, welches sich aus Komponenten und Systemen zusammensetzt.“ (Andresen (2004), S.45)

'The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them. "Externally visible" properties refers to those assumptions other elements can make of an element, such as its provided services, performance characteristics, fault handling, shared resource usage, and so on.' (Software Engineering Institute, nach Bass et al. (2003))

Aus diesen Definitionen können folgende Grundmerkmale der Softwarearchitektur abgeleitet werden: die Architektur beschreibt zum einen die Komponenten eines Softwaresystems und zum anderen deren Beziehungen untereinander und mit der Umwelt. Dabei geht es jedoch nicht um den detaillierten Entwurf des Systems, sondern vielmehr um die Beschreibung der Zusammenhänge zwischen den Anforderungen und dem zu konstruierenden System. Die Wahl einer bestimmten Architektur wirkt sich erheblich auf die späteren qualitativen Eigenschaften des Systems aus. (vgl. Hasselbring (2006), S. 48)

Im Laufe der Zeit haben sich viele verschiedene Architekturmuster herausgebildet und in der Praxis bewährt. Einige Beispiele werden nachstehend kurz aufgeführt (vgl. Hasselbring (2006), S. 48f; Mandl (2008), S. 324ff und Enzyklopädie der Wirtschaftsinformatik (Stand: Juni 2009): <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon>):

- **Monolithische Architektur**

Das IT-System bildet eine untrennbare Einheit, es kann keine Teilung in Subsysteme (Datenhaltung, Verarbeitungslogik, Präsentation) festgestellt werden und alle Funktionen sind eng gekoppelt und zentral implementiert. Dadurch entsteht ein hoher Grad an Komplexität und das System kann nicht die Möglichkeiten zur Austauschbarkeit und Wiederverwendbarkeit bieten, wie sie die moderneren Architekturen aufweisen.

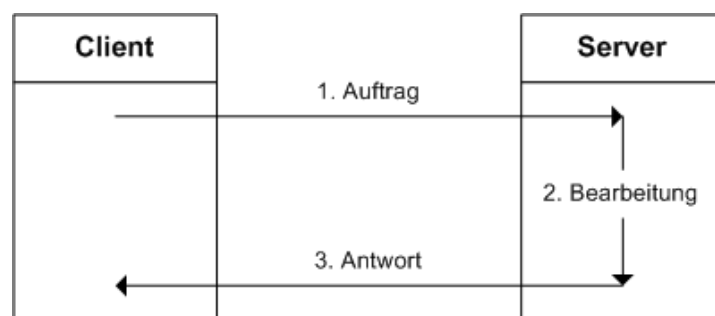
- **Schichtenarchitektur**

Es liegt eine hierarchische Architektur vor, bei der Komponenten einer Schicht immer nur auf die Komponenten darunterliegender Schichten zugreifen können. Dabei wird das System in logische Schichten unterteilt (beispielsweise: Präsentations-, Applikations- und Datenhaltungsschicht) um die Komplexität zu verringern sowie eine Wiederverwendbarkeit der Anwendungsbausteine zu ermöglichen. Die Schichten können dabei auf einem Rechner präsent sein (nur

logische Trennung) oder auf mehrere Geräte verteilt werden (logische und physische Trennung), wodurch eine Dezentralisierung möglich ist.

- **Client/Server Architektur**

Viele Clients (Dienstnutzer) greifen auf einen/wenige Server (Dienstbringer) zu, dabei geht die Kommunikation immer vom Client aus. Eine Schichtenunterteilung ist vorhanden und kann auf unterschiedliche Art und Weise über Client und Server verteilt werden.



Quelle: Enzyklopädie der Wirtschaftsinformatik: Artikel „Client/Server Architektur“
 (<http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/is-management/Systementwicklung/Softwarearchitektur/Architekturparadigmen/Client-Server-Architektur/index.html/?searchterm=architektur>)

Abb. 2.1: Interaktionen bei einer Client/Server-Architektur

- **Peer to Peer Architektur**

Stellt eine Erweiterung des Client/Server-Prinzips dar, wobei nun jede Komponente sowohl als Client als auch als Server agieren kann. Dadurch wird die Skalierbarkeit des Systems verbessert, da mit zunehmender Anzahl an Clients auch die Anzahl der möglichen Server steigt und die Funktionalität auf die Endgeräte (Peers) verlagert wird.

- **Serviceorientierte Architektur**

Funktionalität wird in Form von Diensten bereitgestellt, die über einen Enterprise Service Bus lose gekoppelt sind und über Schnittstellen aufgerufen werden können. Wichtige Grundsätze der Serviceorientierten Architektur sind: Modularität, Granularität, Interoperabilität, Wiederverwendbarkeit, Verwendung offener Standards, Netzwerk-Orientiertheit und Plattformunabhängigkeit.

Es stehen also viele verschiedene Möglichkeiten zur Verfügung ein IT-System zu strukturieren und zu implementieren. Diese Vielzahl an Möglichkeiten führt dazu, dass

sich bei der Durchführung von IT-Projekten häufig die Frage nach der geeigneten Architektur ergibt.

2.2 Bedeutung von Architekturen bei IT-Projekten

Im Folgenden wird des Öfteren von IT-Projekten die Rede sein. Damit sollen alle projektbezogenen Tätigkeiten innerhalb von Unternehmen oder Organisationen gemeint sein, die zur Erschaffung, Erweiterung oder Erneuerung von Informationssystemen dienen, also zum Beispiel die unternehmensweite Einführung von einheitlichen IT-Systemen oder das Konsolidieren verschiedener Datenbestände zu einem System bei Fusionen von Unternehmen. Bei IT-Projekten kommt es in der Praxis häufig zu Problemen, da Zeitaufwand und Kosten den geplanten Rahmen überschreiten oder die Projekte gänzlich aufgrund des Aufwands scheitern. Geeignete IT-Architekturen, innerhalb eines Unternehmens, sind wichtige Faktoren für den Erfolg solcher Projekte (vgl. Herden et al. (2006), S. 1). Hierbei kommt der Architektur besondere Bedeutung zu, da durch ihre konsequente Anwendung und frühzeitige Orientierung an den durch die gewählte Architektur vorgegebenen Systemstrukturen, die Entwicklung des neuen Systems gezielt und geordnet ablaufen kann. Probleme können durch Analyse der zugrunde liegenden Architektur frühzeitig identifiziert und bewältigt werden. Zudem wird durch die Anwendung der geeigneten Architektur erreicht, dass bewährte Lösungen wiederverwendet werden können und sich die entwickelten Komponenten in das System einfügen können. Interoperabilität, Integration und Austauschbarkeit werden durch die einheitlichen Strukturen der Architektur gefördert.

Im Folgenden werden mögliche Beiträge der Softwarearchitektur zur gelungenen Durchführung von IT-Projekten aufgeführt (Andresen (2004), S. 45).

- Die Architektur bildet die konzeptionelle Grundlage und ist ein Rahmenwerk für Veränderungen,
- beschreibt die Struktur eines Systems,
- trägt maßgeblich zur Einheitlichkeit des Systems bei,
- kann ein System sinnvoll unterteilen,
- verfügt über kontextspezifische Sichten und
- betrifft alle Artefakte komponentenbasierter Entwicklung.

2.3 Qualitätsbewertung von Architekturen

2.3.1 Notwendigkeit zur Bewertung der Qualität

In der Softwareentwicklung gibt es Methoden, um den Aufwand und die Kosten für die Entwicklung eines Systems zu messen. So kann über die (Full) Function Point Methode die Anzahl der benötigten Quellcodezeilen ("lines of code") abgeschätzt werden und mit der CoCoMo Methode (Constructive Cost Model) eine Kostenabschätzung vorgenommen werden. Diese Faktoren spielen natürlich bei der Auswahl eines zu entwickelnden Systems eine Rolle. Je nach zugrunde liegender Architektur sind verschiedenste Möglichkeiten für die Entwicklungen von IT-Systemen vorhanden. Doch nur die Kosten für die Entwicklung im Blick zu haben oder eine möglichst schnelle Lösung zu forcieren, kann zur Vernachlässigung der Qualität des Systems führen. Deshalb ist es wichtig, auch die Qualitätsmerkmale des (zukünftigen) IT-Systems stets im Auge zu behalten. Dazu bietet sich die Qualitätsbewertung der Architektur an, da sie maßgeblich die Eigenschaften des entstehenden Systems prägt.

"The earlier you find a problem in a software project, the better off you are. The cost to fix an error found during requirements or early design phases is orders of magnitudes less to correct than the same error found during testing. Architecture is the product of the early design phase, and its effect on the system and the project is profound." (Clements et al. (2002), S. 23)

Probleme, die in den frühen Phasen der Entwicklung gefunden werden, können leichter behoben werden, als solche Probleme, die später (bspw.: in der Testphase) gefunden werden. Die Architekturbewertung ist also ein günstiger Weg, eventuelle Probleme frühzeitig zu beseitigen und sich für die richtige Architektur, für das zu entwickelnde System, zu entscheiden. Außerdem wird laut Clements et al. (2002, S. 37) bei der Durchführung einer Architekturbewertung durch die gemeinsame Betrachtung der Architektur, durch die beteiligten Personen, die Kommunikation untereinander angeregt und das Formulieren einheitlicher Anforderungen und Ziele für das Projekt forciert.

2.3.2 Zeitpunkt der Bewertung

Prinzipiell können Architekturen in jeder Entwicklungsphase bewertet werden, wobei klassischerweise die Bewertung stattfindet wenn die Architektur spezifiziert ist, aber noch nicht mit der Implementierung begonnen wurde. Clements et al. (2002) unterscheiden zwei Ansätze zur Architekturbewertung: früh (early) und spät (late).

- *Früh*: Die Architektur ist noch nicht vollständig spezifiziert und anstehende Architekturentscheidungen sollen durch die Bewertung erleichtert werden.

- *Spät*: Sowohl die Architektur als auch die eigentliche Implementierung sind abgeschlossen, erst dann findet die Bewertung statt. Dieser Fall tritt zum Beispiel auf, wenn Legacy Systeme vorhanden sind und geprüft werden sollen.

2.3.3 Beteiligte Personen

Laut Clements et al. (2002, S. 26) sind zwei wichtige Gruppen an der Qualitätsbewertung der Architektur beteiligt:

- *Bewertungsteam*: Personen, die für die Evaluation und Analyse zuständig sind
- *Stakeholder*: Personen die ein bestimmtes Interesse an der Architektur und dem zu entwickelnden System haben und die über die reine Funktionalität des Systems hinausgehenden Anforderungen repräsentieren (zum Beispiel: Programmierer, Tester, Wartungspersonal)

Eine besondere Art von Stakeholder bildet der Projektentscheider (project decision maker). Dies sind Personen, die ein gesteigertes Interesse an den Ergebnissen der Bewertung haben und Entscheidungen das Projekt betreffend fällen können (bspw.: zur Verfügung stellen von mehr Ressourcen etc.). Dazu gehören: Projektmanager, Architekten, Komponentendesigner.

2.3.4 Qualitätsmerkmale

Eine Softwarearchitektur weist eine ganze Reihe wichtiger Merkmale auf, die untersucht werden können um die Qualität zu bewerten und deshalb während des Entwicklungsprozesses betrachtet werden sollten.

Übersicht über zu bewertende Qualitätsmerkmale (nach Clements et al. (2002), S. 30 ff. und Starke (2008), S. 325 f.):

- *Leistung (Performance)*: Die Leistung eines Systems gibt an wie schnell es auf eingehende Stimuli (Ereignisse/Transaktionen) reagiert und wie viele Ereignisse pro Zeiteinheit verarbeitet werden können. Die Leistung wird in der Regel über Benchmarks gemessen und in der Form *Transaktionen pro Zeiteinheit* oder als *Antwortzeit* (Zeit die das System benötigt um eine bestimmte Transaktion auszuführen) angegeben.

- *Zuverlässigkeit*: Die Fähigkeit eines Systems über längere Zeit fehlerfrei zu laufen, wird als Zuverlässigkeit bezeichnet und durch die *mittlere Zeit bis zu einem Fehler* gemessen.
- *Verfügbarkeit*: Die Verfügbarkeit eines Systems gibt Auskunft über den Zeitanteil, in dem das System läuft und genutzt werden kann. Die Verfügbarkeit wird über die *Zeit zwischen Fehlern* und die *Zeit bis zur Wiederaufnahme der Operationen nach einem Fehlerfall* angegeben.
- *Sicherheit*: Als Sicherheit, wird die Eigenschaft eines Systems bezeichnet, nicht autorisierten Zugriffen zu widerstehen, ohne die Nutzbarkeit für legitime Nutzer einzuschränken.
- *Modifizierbarkeit*: Änderungen (Modifizierungen) an einem System sollten schnell und kosteneffizient umsetzbar sein.
- *Portabilität (Übertragbarkeit)*: Ist ein System portierbar so kann es unter verschiedensten Voraussetzungen (Hard- und Softwareumgebungen) betrieben werden.
- *Funktionalität*: Die Funktionalität eines Systems bezeichnet seine Fähigkeit, die Aufgaben für die es gedacht ist, durch ein koordiniertes Zusammenspiel der Systemkomponenten abzuarbeiten.
- *Variabilität/Flexibilität*: Eine Architektur ist flexibel, wenn es möglich ist sie so zu erweitern oder zu modifizieren, dass eine neue Architektur entsteht, die sich in spezifischer, im Voraus geplanter Art und Weise unterscheidet. Dies ist besonders wichtig, wenn die Architektur als Grundlage für eine ganze Familie von IT-Produkten dienen soll.
- *Unterteilbarkeit*: Die Unterteilbarkeit eines Systems gibt an, ob und wie das System in Teilsysteme aufgeteilt werden kann. Diese Eigenschaft ermöglicht es zum Beispiel im Entwicklungsprozess Teile des Systems bereits lauffähig zu haben, ohne das gesamte System fertigzustellen. Dadurch wird eine inkrementelle Entwicklung möglich.
- *Konzeptuelle Integrität*: Durch konzeptuelle Integrität wird sichergestellt, dass einem System auf allen Ebenen eine einheitliche Vision, einheitliche Themen und Designs zugrunde liegen. Die Architektur soll ähnliche Dinge in ähnlicher Art und Weise erledigen/bewältigen, also konsistent sein.

- *Machbarkeit*: Die Möglichkeit das System mit den vorhandenen Ressourcen (Budget, Zeitplan, Technologie) umzusetzen, muss gegeben sein.

Diese Aufzählung stellt eine Auswahl von Merkmalen dar, über die die Qualität eines Systems bewertet werden kann, erhebt aber keinen Anspruch auf Vollständigkeit.

3 Qualitätsbewertungsmethoden

3.1 Allgemeine Informationen

Im folgenden Kapitel werden verschiedene Methoden vorgestellt, die zur Bewertung von Architekturen genutzt werden können. Einige der Methoden stammen direkt vom Software Engineering Institute (SEI) oder basieren auf dessen Arbeit (SEI-Webseite: http://www.sei.cmu.edu/architecture/evaluation_methods.html). Obwohl mehrere der Methoden inzwischen schon lange existieren und sich vielfach in der Praxis bewährt haben, ist ein einheitlicher Standard für die Bewertung der Qualität von Architekturen im Moment nicht vorhanden. Dies könnte zum Beispiel darin begründet sein, dass in der Praxis die Architekturbewertung, noch immer kein fester Bestandteil des Softwareentwicklungsprozesses ist (vgl. Eicker et al. (2007), S. 1). Obwohl die vorgestellten Methoden häufig unter der Prämisse der Bewertung einer (einzelnen) Softwarearchitektur entwickelt wurden, lässt sich doch zeigen, dass die Methoden genauso auch für die Bewertung mehrerer Architekturen höherer Ebenen (IT-Architekturen usw.) genutzt werden können. Wenn man die Qualitätsmerkmale untersucht, die die Architekturen beeinflussen, wird deutlich dass sie sehr ähnlich sind (vgl. Vasconcelos et al. (2005), S.16).

Einen wichtigen Bestandteil der Bewertungsmethoden bilden Szenarien. Sie dienen dazu, die Qualitätsmerkmale aufzudecken, auf welche die Architektur untersucht werden soll. Sie finden häufig Anwendung, da sie einfach zu kreieren und verstehen sind, keine großen Kosten verursachen und effektiv sind. Szenarien bestehen aus kurzen Anwendungsbeispielen, die eine Interaktion eines Stakeholders mit dem System darstellen, so können sie verschiedene Perspektiven enthalten: Wachstum des Systems, Anwendung des Systems oder explorative (forschende) Systemveränderung. Beispiele für Szenarien wären: Wechsel der Betriebsplattform von Unix zu Macintosh (explorativ), Anwender wechselt ein Graphlayout von horizontal zu vertikal und der Graph wird innerhalb einer Sekunde neu gezeichnet (Anwendung), Verdoppeln der Größe einer existierenden Datenbank ohne die durchschnittliche Antwortzeit zu erhöhen (Wachstum). (vgl. Clements et al. (2002), S. 52 ff.)

3.2 Übersicht der Qualitätsbewertungsmethoden

3.2.1 SAAM

Die Software Architecture Analysis Method (SAAM) ist die erste am SEI entwickelte Bewertungsmethode und auch die erste Methode, die weite Verbreitung und Anwendung fand. Sie war ursprünglich zur Analyse der Modifizierbarkeit eines Softwaresystems gedacht, wurde aber schnell auch zur Evaluation anderer

Qualitätsmerkmale von Architekturen benutzt. Sie gehört (wie viele der folgenden Methoden) zu den Szenario-basierten Qualitätsbewertungsmethoden. (vgl. Clements et al. (2002), S. 211 ff.)

Die folgenden Schritte werden bei einer Evaluation mit der SAAM durchgeführt (vgl. Clements et al. (2002), S. 214 ff.):

- Entwicklung von Szenarien: Alle Stakeholder tragen dazu bei Szenarien zu entwickeln, die möglichst alle Anwendungen des Systems, alle antizipierten Änderungen und (zukünftigen) Qualitätsanforderungen an das System beinhalten.
- Beschreibung der Architektur(en): Die zur Auswahl stehende(n) Architektur(en) werden in einer geeigneten Notation beschrieben, alle Komponenten und relevanten Verbindungen sind ebenso Teil dieser Beschreibung wie das vorhergesagte Verhalten des Systems mit fortlaufender Zeit.

Da sich die ersten beiden Schritte gegenseitig ergänzen (Ideen für Szenarien entstehen bei der Architekturbeschreibung, Szenarien verdeutlichen Anforderungen an die Architektur, welche in die Beschreibung eingehen) werden sie häufig miteinander verknüpft oder in mehreren Iterationen durchgeführt.

- Klassifizierung und Priorisieren der Szenarien: Die entwickelten Szenarien werden in zwei Klassen unterteilt (direkte und indirekte Szenarien) und anschließend durch die Stakeholder nach ihrer Wichtigkeit geordnet.

Direkte Szenarien können durch die vorhandene Architektur gelöst werden, sie betreffen Anforderungen, die bei der Entwicklung der Architektur bereits berücksichtigt wurden. Indirekte Szenarien erfordern dagegen Änderungen oder Ergänzungen an der Architektur, sie helfen insbesondere dabei die Architektur im Hinblick auf den gesamten Lebenszyklus des Systems zu betrachten.

- Individuelle Evaluation indirekter Szenarien: Da bei der SAAM die indirekten Szenarien im Vordergrund stehen (Qualitätsmerkmal: Modifizierbarkeit) werden sie individuell betrachtet, dabei werden die benötigten Änderungen festgehalten und der Aufwand für diese abgeschätzt.
- Bestimmung der Interaktionen zwischen Szenarien: Wenn mehrere Szenarien Änderungen an denselben Komponenten des Systems erfordern, so interagieren sie. Den betroffenen Komponenten sollte besondere Aufmerksamkeit bei der Analyse zugutekommen, da sie sensible Punkte der Architektur darstellen und es

eventuell günstig wäre, Komponenten, die viele Funktionen auf sich vereinigen, abermals in Subkomponenten aufzuteilen.

- Erstellung der Gesamtbewertung: Es wird eine Gewichtung der Szenarien nach ihrem Einfluss auf den Erfolg des Gesamtsystems vorgenommen. Somit wird der Vergleich mehrerer Architekturen, die jeweils unterschiedliche Szenarien unterstützen, erleichtert (Architekturen die bei höher gewichteten Szenarien gut abschneiden sind besser zu bewerten). Und letztendlich wird eine Punktzahl für die positiven und negativen Aspekte der Architektur(en) vergeben und mithilfe der Gewichtung eine Gesamtpunktzahl ermittelt über die dann ein Vergleich verschiedener Architekturen/Architekturvarianten möglich ist.

Eine ausführliche Beschreibung der SAAM ist auf der Webseite des SEI unter http://www.sei.cmu.edu/architecture/scenario_paper/ieee-sw3.htm oder auch bei Clements et al. (2002, S. 211 ff.) zu finden.

3.2.2 ATAM

“The ATAM is an analysis method organized around the idea that architectural styles are the main determiners of architectural quality attributes. The method focuses on the identification of business goals which lead to quality attribute goals. Based upon the quality attribute goals, we use the ATAM to analyze how architectural styles aid in the achievement of these goals.” (Kazman et al. (2000), S. 7)

Die Architecture Tradeoff Analysis Method (ATAM) stellt den Nachfolger der SAAM dar und legt besonderes Augenmerk auf die Identifikation von Geschäftszielen und die damit einhergehenden Qualitätsanforderungen für die Architektur. Die Analyse bezieht dabei auch die Interaktionen zwischen den unterschiedlichen Qualitätszielen mit ein, die eine gezielte Abstimmung (“trade-off“) der Ziele nötig machen.

Eine Qualitätsbewertung nach ATAM läuft folgendermaßen ab (vgl. Clements et al. (2002), S. 44 ff.):

PRÄSENTATION

- Vorstellung der ATAM: Ein Evaluationsleiter stellt die Methode zur Bewertung vor und klärt eventuelle Fragen.
- Vorstellung von Geschäfts-/Projektteilnehmern: Ein Projektverantwortlicher (z.B.: Projektmanager) beschreibt die mit dem Projekt und der anstehenden Entwicklung

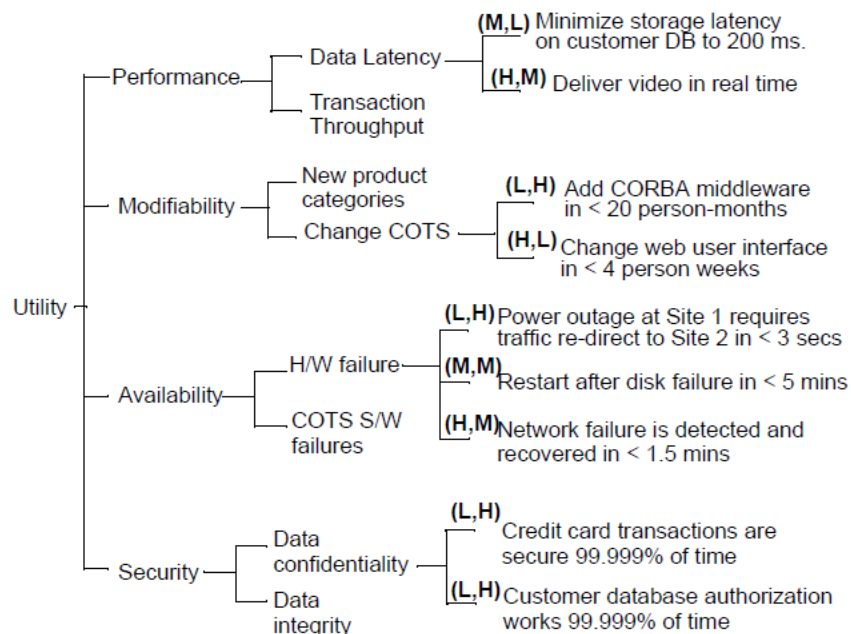
verfolgten Ziele und die Prioritäten bei der Architekturentwicklung (hohe Verfügbarkeit, Sicherheit, schnelle Umsetzbarkeit o.ä.).

- Vorstellung der Architektur: Die vorliegende Architektur wird vorgestellt.

UNTERSUCHUNG UND ANALYSE

- Identifikation der Architekturansätze: Die verfolgte Herangehensweise bei der Architektur wird beschrieben.
- Generierung des “quality attribute utility tree“: Die Qualitätsmerkmale, die später den Nutzen des Systems bestimmen (Performance, Verfügbarkeit, Modifizierbarkeit etc.), werden ausgewählt und zugehörige Szenarien entwickelt und priorisiert.

Nachstehend ein Beispiel für solch einen Nutzenbaum, wobei die Qualitätsmerkmale in Kategorien aufgeteilt werden (Datenlatenz, Hardwarefehler u.a.), die dann verschiedene Szenarien enthalten. Diese Szenarien können dann mit den Angaben High, Medium, Low priorisiert werden, wobei die erste Angabe die Wichtigkeit der betreffenden Eigenschaft angibt und die zweite das Risiko bei der Umsetzung der beschriebenen Anforderung beschreibt.



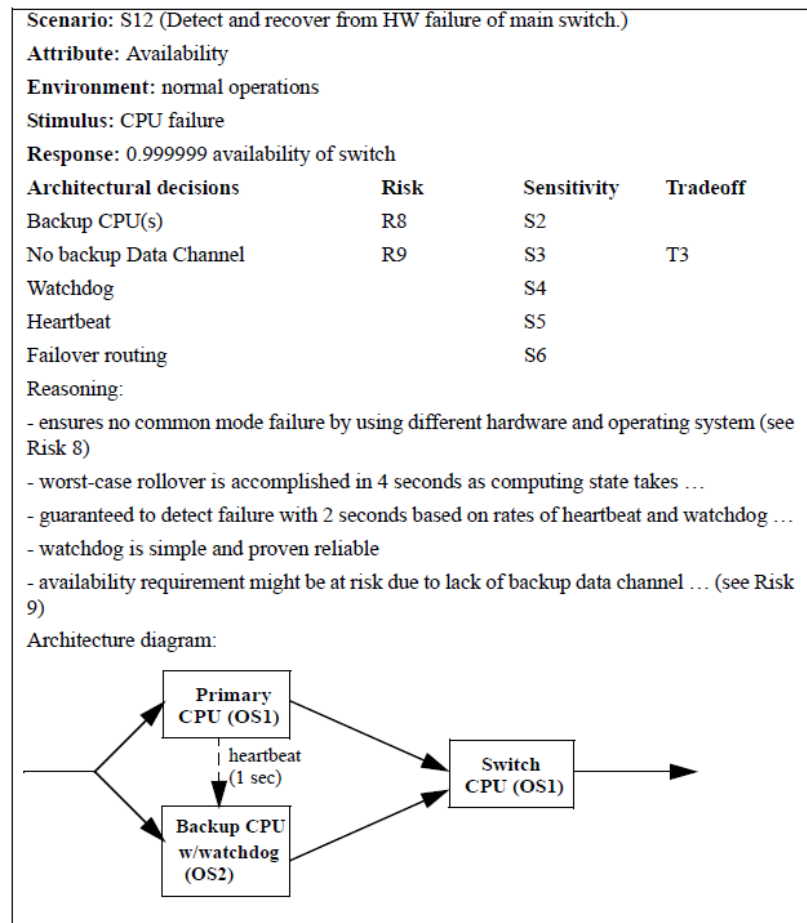
Quelle: Kazman et al. (2000), S. 17

Abb. 3.1: Beispiel für einen Nutzenbaum (“utility tree“)

Analyse der Architekturansätze: Basierend auf den Szenarien mit hoher Priorität werden die Architekturmerkmale analysiert, die für die Erfüllung der Szenarien

nötig sind. Ergebnis dieser Analyse sind: Risiken, Nicht-Risiken, Sensible Punkte der Architektur und Trade-off Möglichkeiten.

Ein Beispiel für eine solche Analyse folgt. Hier ist zu sehen wie ein Szenario (Hardwareausfall) von allen Seiten beleuchtet wird. Die Risiken und Tradeoffs werden erfasst (in separaten Listen gehalten, auf die mit Hilfe der Nummern verwiesen wird).



Quelle: Kazman et al. (2000), S. 32

Abb. 3.2: Beispiel für die Analyse eines Architekturansatzes mit einem Szenario

TESTEN

- Brainstorming und Priorisieren von Szenarien: Eine größere Auswahl an Szenarien (nicht gebunden an den "utility tree") wird von allen Stakeholdern ausgewählt und dann durch Abstimmung erneut priorisiert.
- Analyse der Architekturansätze: Erneute Analyse der Architektur auf Basis der neuen Szenarien, dient zur Kontrolle der bisherigen Ergebnisse, eventuell können so zusätzliche Probleme aufgedeckt werden, die vorher nicht aufgetaucht sind.

BERICHTEN

- Präsentieren der Ergebnisse: Die Ergebnisse werden vom Evaluationsteam vorgestellt.

Mehr Informationen über die ATAM sind auf der Webseite des SEI unter http://www.sei.cmu.edu/architecture/ata_method.html verfügbar.

3.2.3 ARID

Eine Weiterentwicklung der ATAM stellt die Active Reviews for Intermediate Designs Methode (ARID) dar. Sie setzt sich aus Bestandteilen der ATAM und sogenannten Active Design Reviews (ADR) zusammen. Die ADR Methode dient dazu während der Designphase einer Software die Qualität zu überwachen. Dazu wird das Design einer Reihe von Übungen/Aufgaben unterzogen. Die ARID Methode ist für die Evaluation von Teilentwürfen einer Architektur in der frühen Entwicklungsphase gedacht. (vgl. Eicker et al. (2007), S. 4)

Phase 1: Pre-meeting	Step 1: Identify the reviewers
	Step 2: Prepare design presentation
	Step 3: Prepare seed scenarios
	Step 4: Prepare for the review meeting
Phase 2: Review meeting	Step 5: Present ARID method
	Step 6: Present design
	Step 7: Brainstorm and prioritize scenarios
	Step 8: Perform review
	Step 9: Present conclusions

Quelle: Clements (2000), S. 6

Abb. 3.3: Ablauf der ARID Methode

Die obige Abbildung beschreibt den Ablauf der Methode, wobei an Phase 1 nur der Chefentwickler und der Evaluationsleiter beteiligt sind. Das von ihnen bestimmte Review Team führt dann Phase 2 aus. Die genaue Beschreibung der einzelnen Schritte und der Methode an sich ist bei Clements (2000) beschrieben.

3.2.4 QAW

QAW steht für Quality Attribute Workshop, eine weitere von der ATAM abstammende Methode. Diese Methode stellt die Generierung, Priorisierung und Verfeinerung von Qualitäts-Attribut-Szenarien vor der Fertigstellung der Architektur in den Mittelpunkt der Untersuchung. Dies spiegelt sich auch im Ablauf der Methode wieder:

- QAW Präsentation: Die Methode und die Teilnehmer werden vorgestellt.
- Geschäft/Mission vorstellen: Geschäftstreiber hinter dem zu entwickelnden System werden präsentiert.
- Darlegung des Architekturplans: Die vorliegende Architektur wird mithilfe der Dokumentation erläutert.
- Identifikation der Architekturtreiber: Anforderungen, Einschränkungen, Ziele für die Architektur werden identifiziert.
- Szenario Brainstorming: Alle Stakeholder entwickeln Szenarien für ihre jeweiligen Interessen an dem System.
- Szenario Konsolidierung: Ähnliche Szenarien werden zusammengefasst, falls Einigkeit über die Richtigkeit dieser Maßnahme besteht.
- Priorisieren: Die Szenarien werden durch ein Wahlverfahren priorisiert.
- Verfeinern: Die wichtigsten Szenarien werden nach den QAW Vorgaben verfeinert und detailliert untersucht.

Die ausführliche Beschreibung des vorgestellten Ablaufs und der QAW ist bei Barbacci et al. (2003) zu finden.

3.2.5 SAE

Eine weitere Qualitätsbewertungsmethode stellt die Software Architecture Evaluation (SAE) dar, die von der Telekommunikationsfirma AT&T entwickelt wurde. Dies ist eine der ersten Methoden, bei der eine systematische Herangehensweise für die Architekturbewertung vorgeschlagen wurde. Die Basis der Methode bilden Checklisten, mit deren Hilfe untersucht wird, ob eine Softwarearchitektur die an das System gestellten Anforderungen, angemessen und kosteneffektiv erfüllen kann. Durch ihren Ursprung in der Telekommunikationsbranche, liegt bei dieser Methode der Fokus der Bewertung auf den Eigenschaften: Wiederanlauf im Fehlerfall,

Verfügbarkeit, Betrieb, Instandhaltung/Administration und Performance. Die SAE läuft wie folgt ab (vgl. Thiel (2005), S. 59 f.):

- Kontaktaufnahme und Planung: Entwickler und Abteilung, die das Evaluationsverfahren durchführt, organisieren das Review.
- Formung des Evaluationsteams: Der Leiter des Teams sollte sicherstellen, dass Experten für alle Qualitätsmerkmale die untersucht werden sollen, im Team vertreten sind.
- Vorbereitung: Entwickler und Evaluationsteam tragen Informationen für die Bewertung zusammen, woraus die Fragen für das eigentliche Review erarbeitet werden (es entsteht eine Checkliste mit abzuarbeitenden Fragen).
- Analyse: Vorbereitete Fragen werden von den Entwicklern beantwortet.
- Präsentation der Ergebnisse: Die gesammelten Antworten aus der Analyse sowie die zuvor gesammelten Informationen werden genutzt, um einen Bericht über Stärken und Schwächen der Architektur zu erstellen.

3.2.6 AQA

Architecture Quality Assessment (AQA) wurde von der MITRE Corporation entwickelt und basiert auf der Analyse von sechs Qualitätskriterien für Architekturen: Verständlichkeit, Realisierbarkeit, Offenheit, Wartbarkeit, Entwicklungsfähigkeit und Zufriedenstellung des Kunden (vgl. Hillard et al. (1997), S. 4). Bei der Analyse sollen potentielle Risiken einer Architektur identifiziert und Lösungsstrategien entwickelt werden. Sie läuft folgendermaßen ab (vgl. Thiel (2005), S. 61 f.):

- Durchführen einer Anforderungsanalyse: Architekturanforderungen, Ziele und Visionen werden benannt und mit ihnen der weitere Verlauf der Analyse geplant (betrachtete Architekturansätze, Priorisieren, Gewichtung etc.).
- Sammlung von Dokumenten zur Architektur: Alle Informationen, die Architektur betreffend, werden gesammelt.
- Evaluation der Architektur, Messung und Bewertung: Mithilfe der Dokumentation wird die Architektur auf bestimmte Kriterien hin untersucht, die Bewertung erfolgt in vorgegebenen Werten („Ideal, Gut, Marginal, Inakzeptabel, Unvollständig, nicht zutreffend“).

- Interpretation der Ergebnisse und Identifizierung Architekturbedingter Risiken: Alle gesammelten Werte und beantworteten Fragen werden aggregiert, um Punktzahlen für die sechs Qualitätsbereiche zu generieren.
- Dokumentation der Ergebnisse für den Kunden: Die Ergebnisse der Bewertung, gefundene Risiken, offene Fragen usw. werden zusammengefasst und interpretiert.

Die ausführliche Beschreibung der AQA Methode ist in Hillard et al. (1997) dargestellt.

3.2.7 ALMA

Bei der Architecture-Level Modifiability Analysis (ALMA) steht die Modifizierbarkeit des Systems im Vordergrund. Zum Zwecke der Analyse werden verschiedene Analyseziele, explizite Annahmen und gut dokumentierte Analysetechniken verwendet (vgl. Bengtsson et al. (2002), S. 2 ff.). Die Evaluation läuft in fünf Schritten ab, die jedoch nicht streng getrennt sind und mehrmalige Iterationen einzelner Schritte erlauben (vgl. Thiel (2005), S. 63):

- Zielsetzung: Das Ziel der Analyse wird festgelegt, so können beispielsweise eine Abschätzung der Instandhaltungskosten, Risikoanalysen bei Änderungen an der Architektur oder Auswahl einer spezifischen Softwarearchitektur als Ziel dienen.
- Beschreibung der Softwarearchitektur(en): Relevante Aspekte der zur Auswahl stehenden Architekturen werden beschrieben.
- Auflisten von "Change Scenarios": Szenarien zur Veränderung der Architektur werden entwickelt.
- Evaluation: Die Effekte der in den Szenarien beschriebenen Änderungen auf die Architektur werden analysiert.
- Interpretation der Ergebnisse: Die Ergebnisse der Evaluation werden in Hinblick auf die Zielsetzung und die Systemanforderungen interpretiert.

In Bengtsson et al. (2002) ist die Methode ausführlich erläutert.

3.2.8 APSM

Die Instandhaltungsfähigkeit, Wartbarkeit oder "Maintainability" eines Systems ist ein wichtiges Qualitätskriterium, da sie Einfluss auf die Kosten und den Aufwand zum Betreiben des Systems über dessen gesamte Lebensspanne hat. Zur Analyse dieses

Merkmals einer Architektur, bietet sich die Architecture-Level Prediction of Software Maintenance (APSM/ALPSM) an. Sie kann genutzt werden, um verschiedene Architekturen hinsichtlich ihrer Maintainability zu vergleichen oder um dieses Kriterium gegen andere Qualitätsmerkmale der Architektur abzuwiegen (vgl. Bengtsson/Bosch (1999), S. 99ff). Der Ablauf dieser Methode sieht wie folgt aus (vgl. Thiel (2005), S. 64):

- Identifizierung von Kategorien für Wartungsaufgaben: Basierend auf der Domäne und der Applikation des Systems werden Kategorien für Änderungen und Wartungsarbeiten am System bestimmt.
- Synthetisieren von Szenarien: Für jede Kategorie werden repräsentative Szenarien entworfen.
- Gewichtung der Szenarien: Die zuvor festgelegten Szenarien werden nach ihrer Eintrittswahrscheinlichkeit gewichtet.
- Abschätzung der Größe aller Elemente: Das Ausmaß der Architektur, sowie die Größe der einzelnen Komponenten werden zur Abschätzung des Aufwands benötigt.
- Ausarbeitung der Szenarien: Die durch das Szenario ausgelösten Effekte werden bestimmt (Änderungen an Komponenten, neue/entfernte Komponenten) und der verursachte Aufwand wird abgeschätzt.
- Kalkulation des prognostizierten Wartungsaufwands: Mithilfe der zuvor festgelegten Gewichtung und den Ergebnissen der Szenario-Ausarbeitung wird eine Prognose für den Wartungsaufwand des Systems erstellt.

Weitere Informationen zur APSM sind bei Bengtsson/Bosch (1999) zu finden.

3.2.9 PASA

Die Performance Assessment of Software Architectures (PASA) Methode beschäftigt sich explizit mit den Leistungsmerkmalen einer Architektur. Insbesondere wird untersucht, ob Reaktionsgeschwindigkeit und Skalierbarkeit eines Systems durch die vorliegende Architektur unterstützt werden (vgl. Thiel (2005), S. 64). Die Methode ist Szenario-basiert und die folgenden Schritte werden bei der Durchführung abgearbeitet (vgl.: Williams/Smith (2002), S. 2 f.):

- Übersicht über den Evaluationsprozess: Die Gründe, der Ablauf und die zu erwartenden Ergebnisse des Bewertungsverfahrens werden erläutert.
- Architekturbeschreibung: Die Entwickler stellen die vorliegende oder geplante Architektur vor.
- Identifikation von kritischen Anwendungsfällen (“use cases“): Die Eigenschaften und das Verhalten der Architektur, die für die Skalierbarkeit und Reagibilität des Systems entscheidend sind, werden benannt.
- Auswahl von Schlüssel-Szenarien: Für die zuvor benannten Anwendungsfälle werden relevante Szenarien entworfen.
- Identifikation von Leistungszielen: Für die Szenarien werden klare, quantitative, messbare Zielgrößen, die Leistung des Systems betreffend, festgelegt.
- Klärung und Diskussion der Architektur: Die Architektur wird näher betrachtet, Problemzonen werden herausgestellt.
- Architekturanalyse: Es wird untersucht ob, die Architektur die gesetzten Leistungsziele erfüllen kann.
- Auffinden von Alternativen: Sollten Probleme auftreten, wird nach Alternativen Lösungen gesucht, um die angestrebte Leistung zu erzielen.
- Präsentation der Ergebnisse: Ergebnisse und Empfehlungen werden den Managern und Entwicklern mitgeteilt.
- Ökonomische Analyse: Die angefallenen Kosten und erbrachten Leistungen (Architekturverbesserungen) werden analysiert.

Diese Methode wurde von Williams/Smith (2002) entworfen und ausführlich beschrieben.

3.2.10 FAAM

Speziell für ganze Familien von Informationssystemen wurde die Family Architecture Assessment Method (FAAM) entwickelt, welche die strategischen Aspekte und evolutionären Fähigkeiten dieser Systeme besonders betrachtet (vgl. Thiel (2005), S. 65). Der Fokus liegt dabei auf den Qualitätsmerkmalen Interoperabilität und Erweiterbarkeit des Systems. Also auf der Fähigkeit eines Systems, Informationen auszutauschen und auf der Möglichkeit neue Komponenten zu einem System

hinzuzufügen. Die Analyse nach der FAAM gestaltet sich wie folgt (vgl. Dolan (2001), S. 109 ff.):

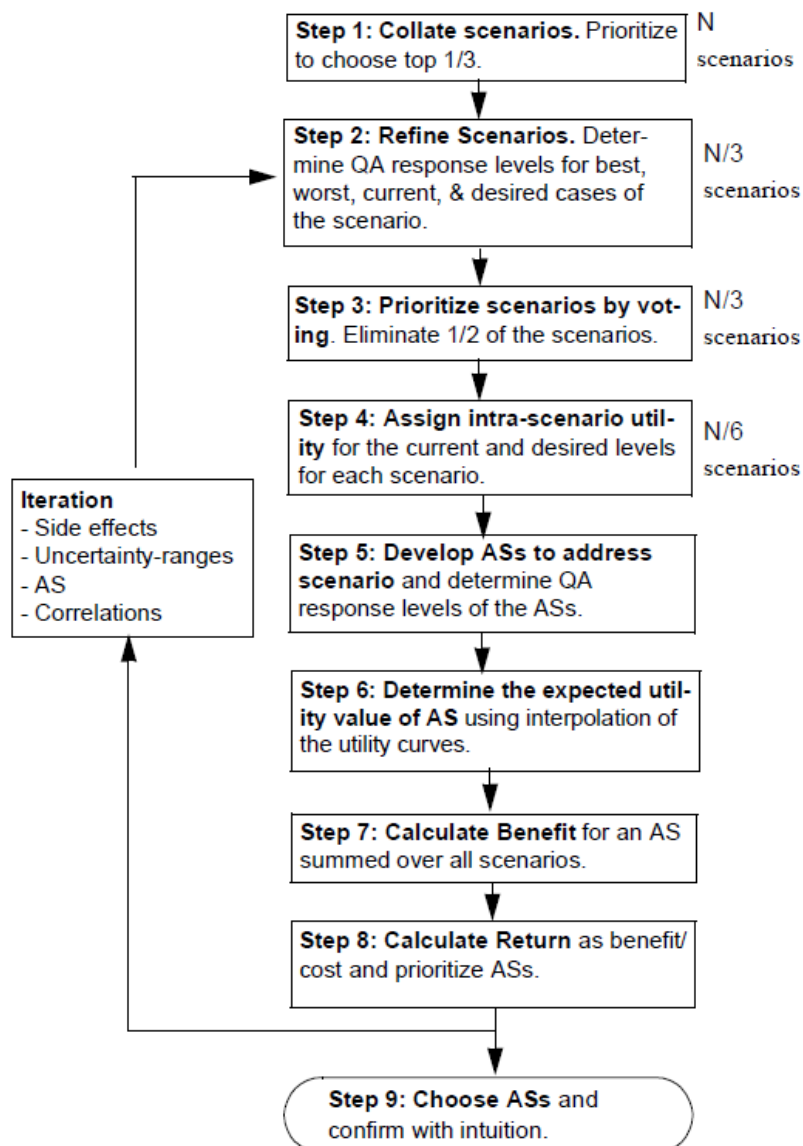
- Definition der Evaluation: Ausmaß und Umfang der Bewertung werden festgelegt, betroffene Systemfamilien benannt und Voraussetzungen adressiert. Daraufhin wird von den Stakeholdern eine Menge von Szenarien die Evolutionen des Systems darstellen (“change cases“) entworfen und priorisiert.
- Vorbereiten von Anforderungen an die Systemqualität: Die Stakeholder entwickeln eine strukturierte Übersicht über Anforderungen, die sie an die Qualität des Systems stellen.
- Vorbereitung der Softwarearchitektur: Die vorliegende Architektur soll so repräsentiert werden, dass ein Vergleich der zuvor erstellten Anforderungen und der betreffenden Architekturmerkmale möglich ist.
- Nachprüfung bisheriger Artefakte: Die Anforderungen, Merkmale und Szenarien werden noch einmal geprüft, um Einigkeit unter den Stakeholdern zu erzielen.
- Bewertung der Konformität der Software Architektur: Die Architektur wird mithilfe der Szenarien analysiert.
- Vorstellung der Ergebnisse und Vorschläge zum weiteren Fortlauf der Entwicklung
- Architekturbewertung ermöglichen: Der Leiter des Bewertungsverfahrens steht den Stakeholdern über die gesamte Dauer des Prozesses unterstützend zur Seite.

Für nähere Informationen zur FAAM wird auf Dolan (2001) verwiesen.

3.2.11 CBAM

Die Cost-Benefit Analysis Method (CBAM) dient zur Analyse der Kosten, der Vorteile und des Einflusses der Architekturwahl auf den Terminplan bei der Entwicklung eines IT-Projekts. Auch die Unsicherheit, die mit den Architekturentscheidungen einhergeht, wird berücksichtigt. (vgl. Ionita et al. (2002), S. 5 f.)

Die folgende Abbildung zeigt den Ablauf der CBAM, hierbei stehen die Abkürzungen AS und QA für Architekturstrategien und Qualitätsattribute. Die Anzahl der betrachteten Szenarien wird stetig verringert und eine zweite Iteration ermöglicht die Einbeziehung weiterer Details.



Quelle: Kazman et al. (2002), S. 8

Abb. 3.4: CBAM Ablauf

Umfassende Informationen zur CBAM und der Durchführung der Methode sind bei Kazman et al. (2002) zu finden.

3.2.12 SAEM

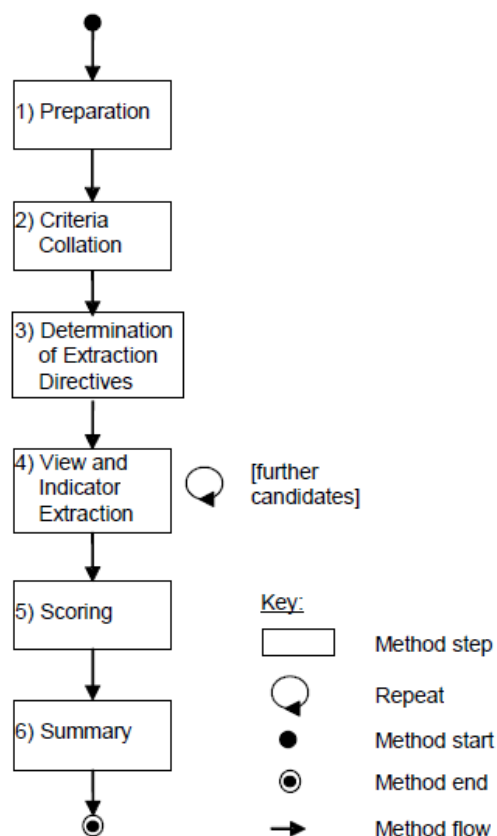
Ein Qualitätsbewertungsmodell, das zum großen Teil auf den aus der Softwaretechnik bekannten Metriken basiert, stellt das Software Architecture Evaluation Model (SAEM) dar. Hierbei wird eine Metrik für Qualitätsattribute nach Vorbild der Goal-Question-Metrik (GQM) entwickelt und geprüft, ob die erzielten Messwerte die angestrebten

Qualitätsmerkmale ermöglichen. Die damit gewonnenen Daten werden zur Vorhersage der Qualität des Systems genutzt. (vgl. Dobrica/Niemelä (2002), S. 647 f.)

Nähere Informationen zum Software Architecture Evaluation Model können bei Dueñas et al. (1999) nachgelesen werden.

3.2.13 SACAM

Zum Vergleich von mehreren Architektur-Kandidaten bei der Systementwicklung, wurde die Software-Architecture Comparison Analysis Method (SACAM) entworfen. Dazu werden bei dieser Methode zwei wichtige Teilziele angestrebt: erstens die Beschreibung der Architekturen durch geeignete Sichten, damit das Feld der Architekturen vergleichbar gemacht wird und zweitens das Zusammentragen von Kriterien, die zum Vergleich der Architekturen dienen sollen.



Quelle: Stoermer et al. (2003), S. 8

Abb. 3.5: SACAM Prozessübersicht

Diese Abbildung zeigt die Schritte der SACAM. Zunächst werden zur Vorbereitung die Geschäftsziele identifiziert und Architekturdokumentationen gesammelt. Dann wird daraus eine Liste von Kriterien abgeleitet und wiederum in Form von Szenarien

festgehalten. Im nächsten Schritt werden Architektursichten gewählt, die es ermöglichen die zuvor gewählten Kriterien bei den vorliegenden Architekturen zu prüfen. Daraufhin werden die Informationen zu den Architekturen (Dokumentation etc.) dazu genutzt, die erstellten Sichten detailliert auszugestalten. Schließlich werden die Architekturen anhand der Architektursicht auf die gewählten Qualitätskriterien hin untersucht und die Ergebnisse, die sich aus der Bearbeitung der Szenarien ergeben, gesammelt. Im letzten Schritt wird eine Zusammenfassung der Ergebnisse erstellt und Empfehlungen für die jeweiligen Architekturen formuliert. Diese und weitere Informationen zur SACAM sind im Bericht von Stoermer et al. (2003) zu finden.

3.2.14 Weitere Qualitätsbewertungsmethoden

Abschließend werden noch einige Methoden genannt, die ebenfalls zum Vergleich herangezogen werden. Ausführliche Beschreibungen der Qualitätsbewertungsmodelle können den angegebenen Quellen entnommen werden.

SAAMCS

Die Software Architecture Analysis Method for Complex Scenarios (SAAMCS) wurde zur Analyse der Flexibilität einer Architektur entwickelt. Dabei werden komplexe Szenarien genutzt, d.h. Szenarien deren Umsetzung in der Realität sich äußerst schwierig gestalten würde, um die Architektur extremen Prüfungen zu unterziehen. (vgl. Lassing et al. (1999), S. 3 ff.)

ESAAMI

Extending SAAM by Integration in the Domain (ESAAMI) oder Integrating SAAM in Domain-centric and Reuse-based Development Processes (ISAAMCR) bezeichnen eine von Molter (1999) vorgestellte Methode zur Evaluation von Architekturen. Im Unterschied zur SAAM werden hier auch Aspekte der Wiederverwendbarkeit und die Domänen-spezifischen Anforderungen an eine Architektur betrachtet.

ASAAM

Mit dem Ziel die Aspekte von Architekturen zu untersuchen, wurde die Aspectual Software Architecture Analysis Method (ASAAM) entwickelt. Mit Aspekten sind dabei Eigenschaften/Verhaltensmerkmale der Architektur gemeint, die sich über viele

Architekturkomponenten erstrecken und aus dem Zusammenspiel dieser ergeben. (vgl. Tekinerdoğan (2004), S. 1 ff.).

SAAMER

Mit der Software Architecture Analysis Method for Evolution and Reusability (SAAMER) können die Qualitätsmerkmale Evolution/Weiterentwicklung und Wiederverwendung einer Architektur untersucht werden. Mehr Details können bei Lung et al. (1997) gefunden werden.

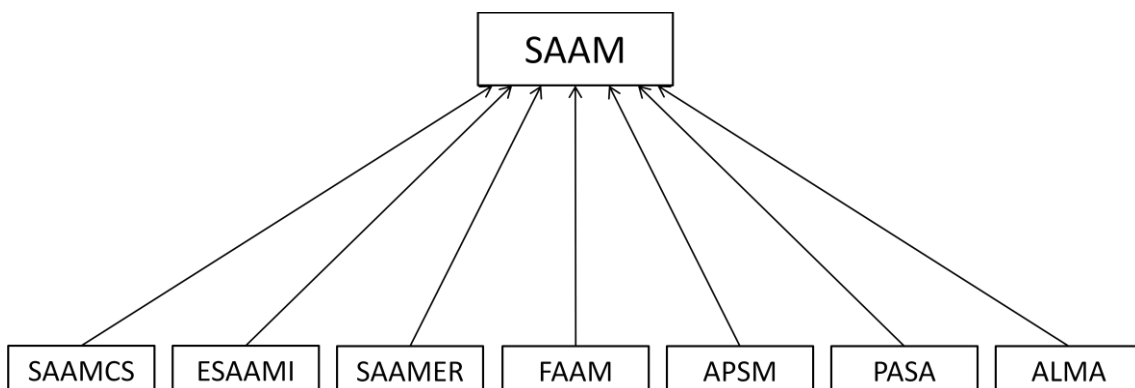
SBAR

Scenario Based Architecture Reengineering (SBAR) bezeichnet eine weitere Qualitätsbewertungsmethode, deren Ziel es ist abzuschätzen, ob eine Architektur das Potential hat, die festgelegten Qualitätsanforderungen zu erfüllen. Diese Methode nutzt neben Szenarien auch Simulationen sowie mathematische Modelle und greift auf Erfahrungswerte der Architekturdesigner zurück. (vgl. Bengtsson/Bosch (1998), S. 2 ff.)

3.3 Ursprung der Methoden

Viele der betrachteten Qualitätsbewertungsmethoden haben ihren Ursprung in der zunächst eigenständig entwickelten SAAM und haben deshalb ähnliche Eigenschaften oder den gleichen Ablauf. Die bestehenden Unterschiede werden später im Vergleich aufgezeigt.

Die folgende Abbildung zeigt die direkt auf der SAAM basierenden Methoden:



Quelle: nach Eicker et al. (2007), S. 4

Abb. 3.6: Spezialisierungen der SAAM

Die ATAM und die SBAR stellen Weiterentwicklungen der SAAM dar, sind aber eigenständige Entwicklungen. Wiederum eine Spezialisierung der ATAM ist die Methode QAW. Die Methoden CBAM, SACAM und ARID bilden eigenständige Weiterentwicklungen der Ideen hinter der ATAM. Vollkommen eigenständige Entwicklungen sind außerdem die SAE, die AQA Methode und die SAEM. (vgl. Eicker et al. (2007), S. 3 ff.)

4 Entwicklung des Kriterienkatalogs für den Vergleich

Für den anstehenden Vergleich wird ein Set von Kriterien benötigt, um die Qualitätsbewertungsmethoden auf Gemeinsamkeiten und Unterschiede hin zu untersuchen. In der Literatur gibt es verschiedene Vorschläge, welche Merkmale der Methoden verglichen werden sollten. So stellen Dobrica und Niemelä (2002, S. 641 ff.) mehrere der oben beschriebenen Methoden anhand ausgewählter Eigenschaften vor und vergleichen sie mittels verschiedener Kriterien. Auch Babar et al. (2004, S. 2 ff.) schlagen ein Rahmenwerk mit zahlreichen Kriterien zum Vergleich der Methoden vor.

Diese beiden Quellen werden von Eicker et al. (2007, S. 6) zur Entwicklung einer Taxonomie zum Vergleich von Evaluationsmethoden herangezogen. Basierend auf diesen Ausführungen wurden die nachfolgenden Kriterien für den durchzuführenden Vergleich ausgewählt.

Untersuchte Qualitätsmerkmale:

Die Anzahl und Art der Qualitätsmerkmale der Architekturen, die bei der jeweiligen Methode betrachtet werden, wird angegeben.

Genutzte Bewertungsansätze:

Die vorgestellten Methoden nutzen unterschiedliche Ansätze zur Bewertung der zu prüfenden Architekturen (Szenarien, Messungen, Checklisten u.a.). Die jeweils genutzten Techniken werden bei diesem Kriterium identifiziert.

Ziel der Methode:

Neben der Vorhersage der Systemqualität durch die Untersuchung der zugrunde liegenden Architektur, können mit einer Architekturbewertung auch andere Ziele verfolgt werden. Einige der Methoden sind für ganz bestimmte Zwecke entwickelt worden, diese Ziele sollen hier bestimmt werden.

Reifegrad/Validierung der Methode:

Bei diesem Kriterium soll untersucht werden inwieweit die Methoden in der Praxis validiert wurden, also bei IT-Projekten (erfolgreich) angewendet worden sind. Außerdem soll eine Aussage über den Entwicklungsstand der vorgestellten Qualitätsbewertungsmethoden getroffen werden.

Anwendungsbereich (Domäne):

Einige der vorliegenden Evaluationsverfahren wurden besonders für ausgewählte Anwendungsbereiche entwickelt (zum Beispiel: Kommunikationsbranche). Ob der Fokus einer Methode auf einer bestimmten Domäne liegt, wird mit diesem Kriterium untersucht.

Stakeholder Beteiligung:

Die Stakeholder werden bei den unterschiedlichen Methoden auf verschiedene Art und Weise beteiligt. Ob Stakeholder in den Prozess einbezogen werden, soll hier geklärt werden.

Ausmaß der Evaluation:

Dieses Kriterium soll Auskunft darüber geben, ob bei der jeweiligen Methode der Vergleich von mehreren Architekturen während eines Evaluationsprozesses explizit ermöglicht wird oder ob die Methode auf die Bewertung einer einzelnen Architektur ausgelegt ist.

Zeitpunkt der Evaluation (Projektphase):

Wie zuvor schon beschrieben, kann eine Qualitätsbewertung zu unterschiedlichen Zeitpunkten während der Entwicklung einer Architektur/eines Systems durchgeführt werden. Hier wird die Projektphase angegeben, für die die jeweiligen Methoden vorgesehen sind.

Voraussetzungen und Hilfsmittel (Input):

In diesem Kriterium sollen alle notwendigen Bedingungen und Voraussetzungen für die Anwendung eines Verfahrens erfasst werden. Dies könnten notwendige Dokumente (Architekturbeschreibung o.ä.) oder auch organisatorische oder ressourcenabhängige Anforderungen sein. Des Weiteren werden, falls vorhanden, die angegebenen Hilfsmittel für die jeweiligen Evaluationsverfahren festgehalten.

Ergebnisse (Output):

Hier werden die Ergebnisse des Evaluationsverfahrens und die damit verbundenen Leistungen/Vorzüge, die durch das anwenden der Methode entstehen, angegeben.

Detaillierungsgrad der Prozessbeschreibung:

Dieses Kriterium beschreibt die Art und Weise wie die Vorgehensweise der Methode in der Literatur beschrieben ist. Eine ausführliche und detaillierte Beschreibung der Prozessschritte sowie klare Formulierungen zu den Bedingungen und Richtlinien einer Methode helfen dabei, die Evaluation erfolgreich und gezielt durchzuführen.

Benötigte Ressourcen:

Alle für die Durchführung eines Qualitätsbewertungsverfahrens notwendigen Ressourcen, sollen in diesem Kriterium erfasst werden, im Wesentlichen wird es sich dabei um Zeit- und Personenaufwand (Größe des Evaluationsteams) handeln.

Benötigte Kenntnisse und Erfahrungen des Evaluationsteams:

Falls besondere Kenntnisse des Evaluationsteams für eine bestimmte Methode erforderlich sind oder ein gewisser Grad an Erfahrung von Nöten ist, so wird dies in diesem Kriterium zum Ausdruck gebracht.

5 Vergleich der Qualitätsbewertungsmethoden

Nun wird der Vergleich der in den vorhergehenden Kapiteln vorgestellten Architektur-Bewertungsmethoden durchgeführt. Dabei wird zunächst eine tabellarische Darstellung erstellt, in der die Angaben für die jeweiligen Methoden zu den vorhergehend bestimmten Kriterien aufgeführt sind. Anschließend erfolgt zu jedem der Kriterien eine zusammenfassende Betrachtung, um die vorhandenen Gemeinsamkeiten und Unterschiede hervorzuheben.

Da die folgende Vergleichstabelle sehr umfangreich ist, wird sie über mehrere Seiten aufgeteilt. Es werden pro Seite drei der Evaluationsmethoden mit den jeweiligen Daten zu den Kriterien dargestellt. Zur besseren Übersicht folgt nachstehend ein kurzes Verzeichnis mit den Methoden und dem Teil der Tabelle, in dem sich die zugehörigen Angaben befinden:

- SAAM, ATAM, ARID: *Tabelle 5.1*
- QAW, SAE, AQA: *Tabelle 5.2*
- ALMA, APSM, PASA: *Tabelle 5.3*
- FAAM, CBAM, SAEM: *Tabelle 5.4*
- SACAM, SAAMCS, ESAAMI: *Tabelle 5.5*
- ASAAM, SAAMER, SBAR: *Tabelle 5.6*

Als Quelle für die in der Tabelle zusammengefassten Angaben dienen die Ursprungsdokumente/Beschreibungen der Qualitätsbewertungsmethoden oder die zuvor schon genannten Arbeiten von Dobrica und Niemelä (2002, 651 f.), Babar et al. (2004, S. 8) sowie Eicker et al. (2007, S. 7 f.).

Methode	SAAM	ATAM	ARID
Untersuchte Qualitätsmerkmale	Modifizierbarkeit	Diverse Qualitätsmerkmale	Eignung/Tauglichkeit des Architekturdesigns
Genutzte Bewertungsansätze	Szenarien	Utility-tree, Szenarien, Fragekatalog	Szenarien, Active Design Reviews
Reifegrad/Validierung der Methode	Ausgereift, vielfach in vielen Bereichen angewendet	Ausgereift, überarbeitet, häufig angewendet, Fallstudien und Trainings verfügbar	Verwendet validierte Grundlagen (ATAM und ADR), aber noch keine eigene Validierung
Stakeholder Beteiligung	Umfangreiche Beteiligung aller Stakeholder	Repräsentation/Anwesenheit aller Stakeholder zur Evaluation nötig	Stakeholder selbst werden als Reviewer genutzt
Ziel der Methode	Feststellung der Architekturqualität bezüglich der Modifizierbarkeit	Auffinden von Stärken und Schwächen einer Architektur	Evaluation des Architekturdesigns hinsichtlich der Brauchbarkeit
Detaillierungsgrad der Prozessbeschreibung	Sehr ausführliche Schritt-für-Schritt Beschreibung inkl. Fallstudie	Sehr ausführliche Schritt-für-Schritt Beschreibung inkl. Fallstudie	Sehr ausführliche Schritt-für-Schritt Beschreibung inkl. Fallstudie
Voraussetzungen (V) und Hilfsmittel (H)	V: Architekturbeschreibung	V: Architekturentwurf, Anforderungsanalyse H: Utility-tree	V: Architekturdesign H: ADR
Zeitpunkt der Evaluation (Projektphase)	Zum Ende des Architekturdesigns	Nach Abschluss der Entwurfsphase oder als iterativer Verbesserungsprozess beim Entwurf	Während des Komponentendesigns
Benötigte Kenntnisse und Erfahrung des Evaluationsteams	Leicht anzuwendende Methode, kaum Vorkenntnisse nötig	Umfangreiche Kenntnisse (u.a.) zum Aufstellen des Utility-tree nötig	Kenntnisse zu ADR und ATAM nötig
Benötigte Ressourcen	Variabel, Durchführung in zwei Tagen möglich, Evaluationsteam (3 Pers.)	Mindestens 3 Tage nötig, Evaluationsteam (4 Pers.)	2 Tage Mindestdauer, Evaluationsteam (2 Pers.)
Ergebnisse	Szenarien zur Repräsentation von zukünftigen Änderungen, Aufwandsabschätzungen	Risiken, Nicht-Risiken, Tradeoff- und Sensitivity-Points	Aussagen über Tauglichkeit des Designs
Anwendungsbereich (Domäne)	Keine Einschränkung	Keine Einschränkung	Keine Einschränkung
Ausmaß der Evaluation	Eine einzelne Architektur oder mehrere Architekturen	Eine einzelne Architektur	Eine einzelne Architektur

Tabelle 5.1: Vergleich der Qualitätsbewertungsmethoden (Teil 1)

Methode	QAW	SAE	AQA
Untersuchte Qualitätsmerkmale	Diverse Qualitätsmerkmale	Fehlerbehebung, Wartung, Operation, Administration, Leistung, Kosten	Verständlichkeit, Machbarkeit, Offenheit, Instandhaltung, Entwicklungsfähigkeit, Kundenzufriedenheit
Genutzte Bewertungsansätze	Szenarien	Szenarien	Fragekatalog/ Befragungstechnik, Messungen
Reifegrad/Validierung der Methode	Validierung angelaufen, Anwendung erfolgt	Keine Angaben	Prototyp, in Projekten der Air Force und amerik. Flugaufsicht genutzt
Stakeholder Beteiligung	Umfangreiche Stakeholderbeteiligung (5 - 30 Teilnehmer)	Projektmanagement, Entwickler und Architekten beteiligt	Keine Beteiligung
Ziel der Methode	Identifizierung wichtiger Qualitätsanforderungen, Analyse der gefundenen Qualitätsmerkmale	Auffinden von Stärken und Schwächen einer Architektur	Unterstützung bei der Entscheidungsfindung durch Architekturanalyse
Detailierungsgrad der Prozessbeschreibung	Ausführliche Schritt-für-Schritt Beschreibung	Kurze Aufzählung der Prozessschritte	Nur Konzeptbeschreibung
Voraussetzungen (V) und Hilfsmittel (H)	V: Architekturentwurf/-plan H: Templates/Beispiele für Szenarien	V: Architekturentwurf	V: Architekturbeschreibung, -Entwurf, Anforderungsanalyse H: Scoring-Tabelle
Zeitpunkt der Evaluation (Projektphase)	Vor oder während der Festlegung des Architekturdessigns	Nach dem Architekturentwurf	Vor der Implementierung des Systems
Benötigte Kenntnisse und Erfahrung des Evaluationsteams	Keine Angaben	Keine Angaben	Kenntnisse für Messungen/Befragungen nötig
Benötigte Ressourcen	Keine Angaben	Meeting über 2 Tage	Keine Angaben
Ergebnisse	Architekturtreiber, verfeinerte und priorisierte Szenarien	Aufgedeckte Schwächen/Unstimmigkeiten in der Architektur	Zusammenfassung und Interpretation der Evaluation, Liste von offenen Punkten
Anwendungsbereich (Domäne)	Keine Einschränkung	Telekommunikationssysteme	Keine Einschränkung
Ausmaß der Evaluation	Eine einzelne Architektur	Eine einzelne Architektur oder mehrere Architekturen	Eine einzelne Architektur oder mehrere Architekturen

Tabelle 5.2: Vergleich der Qualitätsbewertungsmethoden (Teil 2)

Methode	ALMA	APSM	PASA
Untersuchte Qualitätsmerkmale	Modifizierbarkeit	Wartung/Instandhaltung	Leistung (Performance)
Genutzte Bewertungsansätze	Szenarien	Szenarien	Szenarien, Modelle, Messungen/ Simulationen
Reifegrad/Validierung der Methode	Mehrfach erfolgreich angewendet	Bei mindestens zwei Systemen erfolgreich genutzt	In verschiedenen Domänen bereits validiert
Stakeholder Beteiligung	Stakeholder sind beteiligt (meist in Form von Interviews)	Nur Beteiligung des Designers	Manager und Entwickler sind beteiligt
Ziel der Methode	Analyse der Auswirkungen von Änderungen auf die Architektur	Abschätzung des Instandhaltungsaufwands	Feststellen der Leistungsmerkmale einer Architektur
Detailierungsgrad der Prozessbeschreibung	Ausführliche Prozessbeschreibung mit Fallstudie	Ausführliche Beschreibung der Prozessschritte mit Fallstudie	Allgemeine Prozessbeschreibung mit Anwendungsbeispiel
Voraussetzungen (V) und Hilfsmittel (H)	V: Architekturbeschreibung	V: Architekturentwurf, Anforderungsanalyse H: Berechnungsformel	V: weit fortgeschrittener Architekturentwurf (zur Erstellung von Modellen)
Zeitpunkt der Evaluation (Projektphase)	Während der Designphase	Während der Designphase	Spätestens vor der Implementierung
Benötigte Kenntnisse und Erfahrung des Evaluationsteams	Keine Angaben	Keine Angaben	Keine Angaben
Benötigte Ressourcen	Keine Angaben	Keine Angaben	Keine Angaben
Ergebnisse	Szenarien mit Analyse der Auswirkungen, je nach Zielsetzung	Abschätzung des Instandhaltungsaufwands	Performanceanalyse mit Verbesserungsvorschlägen, Kostenanalyse
Anwendungsbereich (Domäne)	Keine Einschränkung	Keine Einschränkung	Keine Einschränkung
Ausmaß der Evaluation	Eine einzelne Architektur oder mehrere Architekturen	Eine einzelne Architektur oder mehrere Architekturen	Eine einzelne Architektur

Tabelle 5.3: Vergleich der Qualitätsbewertungsmethoden (Teil 3)

Methode	FAAM	CBAM	SAEM
Untersuchte Qualitätsmerkmale	Interoperabilität, Erweiterbarkeit	Kosten, Nutzen, Risiken, Zeitplan	Externe und interne Qualitätsanforderungen
Genutzte Bewertungsansätze	Szenarien	Szenarien, Wertbestimmung für Szenarien, Kostenmodell, ROI	Metriken, GQM
Reifegrad/Validierung der Methode	Bei verschiedenen Informationssystemen angewendet	Ausgereift, in der Praxis angewendet, weiterentwickelt	Keine Angaben
Stakeholder Beteiligung	Hauptsächlich Stakeholder mit strategischen Interessen	Stakeholder aus verschiedenen Bereichen sind beteiligt	Keine Beteiligung
Ziel der Methode	Identifikation von Stärken und Schwächen, Verbesserungsvorschläge abgeben	Betrachtung der wirtschaftlichen Auswirkungen von Entwurfsentscheidungen	Vorhersage der Systemqualität
Detailierungsgrad der Prozessbeschreibung	Beschreibung der Prozessschritte mit Besonderheiten der Methode und Fallstudie	Kurze Schritt-für-Schritt Beschreibung und Fallstudie	Nur grobe Beschreibung des Konzepts
Voraussetzungen (V) und Hilfsmittel (H)	V: Architekturentwurf H: verschiedene Templates	V: ATAM Analyse H: Grundlagen zu ROI und Unsicherheit	V: System muss (teilweise) implementiert sein (für Messungen)
Zeitpunkt der Evaluation (Projektphase)	Nach der Architekturspezifikation	Nach/Während der Entwurfsphase	Relativ spät während des Entwicklungsprozesses
Benötigte Kenntnisse und Erfahrung des Evaluationsteams	Keine Angabe	Keine Angabe	Wissen über Qualität von Software und Metriken nötig
Benötigte Ressourcen	3 - 4 Tage, Evaluationsteam variabler Größe	Keine Angabe	Keine Angaben
Ergebnisse	Analyseresultate und Verbesserungsvorschläge	Risiken und Architekturstrategien, berechnete Werte (ROI)	Messergebnisse
Anwendungsbereich (Domäne)	Informationssysteme/-systemfamilien	Keine Einschränkung	Keine Einschränkung
Ausmaß der Evaluation	Eine einzelne Architektur oder mehrere Architekturen	Eine einzelne Architektur oder mehrere Architekturen	Eine einzelne Architektur

Tabelle 5.4: Vergleich der Qualitätsbewertungsmethoden (Teil 4)

Methode	SACAM	SAAMCS	ESAAMI
Untersuchte Qualitätsmerkmale	Diverse (abhängig vom Geschäftsziel)	Flexibilität	Wiederverwendbarkeit, Flexibilität
Genutzte Bewertungsansätze	Szenarien	(Komplexe) Szenarien	Szenarien
Reifegrad/Validierung der Methode	Laut Entwickler bereits erfolgreich angewendet	Keine Angaben	Keine Angaben
Stakeholder Beteiligung	Stakeholder sind in verschiedenen Rollen beteiligt	Stakeholderbeteiligung wird angedeutet	Alle Stakeholder beteiligt
Ziel der Methode	Unterstützung des Architekturauswahlprozesses durch Vergleich mehrerer Architekturen	Untersuchung der Komplexität von Szenarien und deren Effekt auf die Architektur	Feststellen der Wiederverwendbarkeit einer Architektur durch Flexibilitätsanalyse
Detailierungsgrad der Prozessbeschreibung	Ausführliche Schritt-für-Schritt Beschreibung mit Beispielen	Kurze Konzeptbeschreibung	Konzeptbeschreibung anhand der Erweiterung der SAAM
Voraussetzungen (V) und Hilfsmittel (H)	V: mehrere Architekturkandidaten, Identifikation der Geschäftsziele	V: Architekturentwurf	V: Architekturentwurf H: Template (angedeutet)
Zeitpunkt der Evaluation (Projektphase)	Vor der Architekturauswahl	Vor der Implementierung	Zum Abschluss des Entwurfsprozesses
Benötigte Kenntnisse und Erfahrung des Evaluationsteams	Wirtschaftl. Kenntnisse nötig um Geschäftsziele auf Qualitätsmerkmale abzubilden	Kenntnis der SAAM vorausgesetzt	Keine Angaben
Benötigte Ressourcen	Workshop (ca. 3 Tage) + 1,5 Wochen Bearbeitung durch Evaluationsteam	Keine Angaben	Keine Angaben
Ergebnisse	Report mit Bewertungen (Scores) für die Architekturkandidaten	Liste von Szenarien mit Bewertung der Auswirkungen	Evaluierte, gewichtete Szenarien, Abschätzung der Wiederverwendbarkeit
Anwendungsbereich (Domäne)	Keine Einschränkung	Keine Einschränkung	Keine Einschränkung
Ausmaß der Evaluation	Untersucht immer mehrere Architekturkandidaten gleichzeitig	Eine einzelne Architektur oder mehrere Architekturen	Eine einzelne Architektur oder mehrere Architekturen

Tabelle 5.5: Vergleich der Qualitätsbewertungsmethoden (Teil 5)

Methode	ASAAM	SAAMER	SBAR
Untersuchte Qualitätsmerkmale	Diverse (je nach betrachtetem Aspekt)	Wiederverwendung, Weiterentwicklung (Evolution)	Diverse Qualitätsmerkmale
Genutzte Bewertungsansätze	Szenarien	Szenarien	Szenarien, Simulation, mathematische Modelle, Erfahrungen
Reifegrad/Validierung der Methode	Keine Angaben	In mindestens einem Projekt genutzt	Keine Angaben zur Weiterentwicklung oder Validierung
Stakeholder Beteiligung	Stakeholder nur als Szenariolieferanten	Stakeholder sind beteiligt	Nur Architekturdesigner ist beteiligt
Ziel der Methode	Identifikation und Analyse von Aspekten der Architektur	Analyse der Evolutionsfähigkeit und Wiederverwendbarkeit der Architektur	Bestimmung des Potenzials zur Erreichung der Anforderungen
Detaillierungsgrad der Prozessbeschreibung	Nur Konzeptbeschreibung	Nur Konzeptbeschreibung	Konzeptbeschreibung mit Beispiel
Voraussetzungen (V) und Hilfsmittel (H)	V: Architekturentwurf H: Heuristische Regeln	V: Implementiertes System	V: Implementiertes System, Anforderungsanalyse
Zeitpunkt der Evaluation (Projektphase)	Früh im Entwicklungsprozess	Spät, nach der Implementierung	Spät, nach der Implementierung
Benötigte Kenntnisse und Erfahrung des Evaluationsteams	Keine Angaben	Keine Angaben	Relativ hohe Anforderungen aufgrund verschiedener Ansätze
Benötigte Ressourcen	Keine Angaben	Keine Angaben	Keine Angaben
Ergebnisse	Identifikation der Aspekte der Architektur	Szenarien mit Auswirkungsanalyse	Analyse und Reengineering der Architektur
Anwendungsbereich (Domäne)	Keine Einschränkung	Keine Einschränkung	Keine Einschränkung
Ausmaß der Evaluation	Eine einzelne Architektur	Eine einzelne Architektur	Eine einzelne Architektur

Tabelle 5.6: Vergleich der Qualitätsbewertungsmethoden (Teil 6)

Untersuchte Qualitätsmerkmale

Während die vorgestellten Qualitätsbewertungsmethoden alle einen gemeinsamen Untersuchungsgegenstand in der IT-Architektur haben, unterscheiden sie sich doch darin, welches Qualitätsmerkmal untersucht werden soll. Zunächst ist zu unterscheiden, ob nur ein/einige Qualitätsmerkmale geprüft werden oder ob diverse Merkmale zur Untersuchung stehen. So bleibt festzuhalten, dass die ATAM, QAW, SAEM, SACAM, ASAAM und SBAR die zu untersuchenden Merkmale nicht einschränken, also für die Bewertung einer Vielzahl von Qualitätsmerkmalen geeignet sind. Demgegenüber sind die SAAM und die restlichen Methoden auf bestimmte Qualitätsmerkmale beschränkt. Hier zeigt sich dann, dass der Fokus der Methoden auf verschiedene Attribute gelegt wird. Im Vordergrund stehen die Modifizierbarkeit (SAAM, ALMA), Wartung und Instandhaltung (APSM), Leistung (PASA), Flexibilität/Wiederverwendbarkeit (SAAMCS, ESAAMI) oder weitere Sets von Qualitätsattributen. Und auch die Kosten werden bei einigen Methoden berücksichtigt (CBAM). Die untersuchten Qualitätsmerkmale sind ein großer Unterscheidungsfaktor zwischen den Bewertungsmethoden. Bei den Methoden, die sich nicht von vornherein auf bestimmte Merkmale festlegen, können dann auch noch von Anwendung zu Anwendung unterschiedliche Merkmale im Vordergrund stehen, je nach Geschäftsziel, Interesse der Stakeholder oder Ziel der Evaluation.

Genutzte Bewertungsansätze

Bei fast allen Methoden kommen Szenarien als Teil des Bewertungsansatzes zum Einsatz. Dies ist mit der simplen Anwendung und der Flexibilität von Szenarien zu erklären. Außerdem sind Szenarien nicht auf bereits vorhandene Implementierungen angewiesen, sondern können als rein theoretische Gedankenspiele angesehen werden, mit denen die Architektur aus verschiedenen Perspektiven untersucht werden kann. Sie sind darüber hinaus kostengünstig und haben sich in der Praxis als effektiv bewährt. Viele Methoden nutzen zusätzlich zu den Szenarien weitere Techniken zur Bewertung. So werden zum Beispiel bei der ATAM so genannte Utility-trees („Nutzenbäume“) genutzt um, die Evaluation durchzuführen und bei der ARID kommen Active Design Reviews zum Einsatz. Weitere Techniken sind Befragungen oder Fragenkataloge (ATAM, AQA), Simulationen (PASA, SBAR) oder Messungen und Metriken (SAEM, AQA, PASA) sowie mathematische Modelle (SBAR). Zu den Methoden die keinen Einsatz von Szenarien vorsehen, gehören die AQA und die SAEM. Szenarien stellen also mit wenigen Ausnahmen einen gemeinsamen Bestandteil der Methoden dar.

Reifegrad/Validierung der Methode

Die Angaben zur Validierung oder zum Reifegrad der verschiedenen Methoden sind teilweise nur vage oder gar nicht vorhanden. Einige Sachverhalte lassen sich aber ableiten. Zumindest die SAAM, ATAM und CBAM sind als ausgereift zu bezeichnen und haben sich laut Aussagen der Entwickler der Verfahren (durch Erfahrungen als Teil von Evaluationsteams) in der Praxis bewährt. Diese Methoden haben auch schon Überarbeitungen durchgemacht oder unterliegen einer stetigen Weiterentwicklung. Besonders zur ATAM gibt es eine Fülle von Fallstudien und Trainings, die die weitere Verbreitung der Methode fördern. Andere Methoden wurden bereits erfolgreich angewendet, dazu gehören: QAW, AQA, ALMA, APSM, PASA, FAAM, SACAM und SAAMER. Zu den übrigen Qualitätsbewertungsmethoden wurden keine Informationen zum Entwicklungs-/Anwendungsstand gefunden.

Stakeholder Beteiligung

Die Interessen der Stakeholder haben für die Bewertung von Architekturen eine hohe Bedeutung, deshalb sind sie häufig an dem Evaluationsverfahren selbst beteiligt. Nur bei wenigen Methoden wird komplett auf den Einsatz von Anspruchsgruppen verzichtet (AQA, SAEM) oder die Beteiligung rein auf den Designer der Architektur beschränkt (APSM, SBAR). Der Großteil der Evaluationsverfahren nutzt die Stakeholder in verschiedenster Form, vom reinen Interviewpartner ALMA, über die Beteiligung zur Entwicklung von Szenarien (ASAAM) bis hin zur umfangreichen Teilnahme am gesamten Evaluationsverfahren (SAAM, ATAM) oder gar der mehr oder weniger selbstständigen Durchführung des Verfahrens (ARID). Dabei ist auch die Rolle des Evaluationsteams zu beachten, das entweder tatsächlich die Evaluation alleine durchführt, mit den Stakeholdern zusammen den Prozess absolviert oder nur in der Rolle des Moderators ('facilitator') auftritt. Hier sind also tatsächlich von Methode zu Methode unterschiedlichste Auffassungen vertreten.

Ziel der Methode

Wie bereits erwähnt haben alle Methoden die Untersuchung der Qualität einer Architektur, durch die Analyse der Qualitätsmerkmale der Architektur, als Ziel gemeinsam. Doch die Ausprägungen dieses Grundvorhabens aller Methoden fallen durchaus unterschiedlich aus. In einigen Fällen wird das Ziel etwas eingeschränkt, so dient beispielsweise die SAAM zur Feststellung der Architekturqualität bezüglich der Modifizierbarkeit, es wird also eine Beschränkung auf nur einen Aspekt der Qualität vorgenommen. Dies ist auch logisch, denn wenn nur ein bestimmtes Qualitätsattribut

untersucht wird, kann das Ziel auch nur sein, die Qualität bezüglich eben dieses Kriteriums zu beurteilen. Weitere Ziele der Methoden können sein: Auffinden von Stärken und Schwächen einer Architektur (ATAM, SAE, FAAM), Unterstützung bei der Entscheidungsfindung/Architekturauswahl (SACAM, AQA) oder auch die Identifizierung wichtiger Qualitätsanforderungen (QAW). Zusätzlich wird bei einigen der Qualitätsbewertungsmethoden auch die Untersuchung der Kosten und des Aufwands als Folge von Architekturentscheidungen als Ziel angegeben (CBAM, APSM).

Detailierungsgrad der Prozessbeschreibung

Bei der Beschreibung der vorliegenden Bewertungsmethoden gibt es zum Teil große Unterschiede. Während einige Verfahren ausführlich und detailliert Schritt-für-Schritt erläutert werden, liegen zu anderen Methoden nur vage Konzepte vor. Zu den sehr gut beschriebenen Methoden zählen die SAAM und die ATAM, die nicht nur in verschiedenen Papers/Artikeln der SEI vorgestellt werden, sondern auch in Büchern zu dem Thema genau beschrieben werden. Hier wird jeder Schritt des Prozesses erläutert und der Gesamtablauf geschildert. Hinzu kommen Fallstudien zur Verdeutlichung der Anwendung der jeweiligen Methode in der Praxis. Zu den ebenfalls gut dokumentierten Methoden zählen: ARID, QAW, ALMA, APSM, FAAM, und SACAM. Kürzere allgemeinere Beschreibungen der Prozesse liegen für die SAE, PASA, CBAM und SBAR vor. Für die restlichen Methoden sind nur grobe Konzeptbeschreibungen ohne Beispiele oder Fallstudien vorhanden. Da sich viele dieser Methoden jedoch an der SAAM orientieren oder von ihr abstammen, kann angenommen werden, dass der Ablauf dieser Methoden dem SAAM-Prozess ähnelt. Deshalb macht eine kürzere Beschreibung des Konzeptes anhand einer Erweiterung der SAAM wie zum Beispiel bei der ESAAMI durchaus Sinn. Trotzdem bleibt festzuhalten, dass mit steigender Detailliertheit der Methodenbeschreibung die Anwendung erleichtert werden sollte.

Voraussetzungen und Hilfsmittel (Input)

Die Voraussetzungen für die Durchführung eines Evaluationsverfahrens ähneln sich bei den meisten Methoden stark. Meist wird ein Architekturentwurf oder zumindest eine Beschreibung oder ein Plan für die vorgesehene Architektur benötigt. Es werden also logischerweise für die Bewertung einer Architektur hinsichtlich verschiedener Qualitätsmerkmale auf jeden Fall Informationen über die Architektur benötigt. Hinzu kommt bei einigen Methoden die Forderung nach einer vorhandenen Anforderungsanalyse (ATAM, AQA, APSM) oder im Falle der SACAM die Identifikation von Geschäftszielen. Bei der SACAM gibt es eine weitere Besonderheit, denn hier sind

explizit mehrere Architekturkandidaten vorausgesetzt, um sie zu vergleichen. Die CBAM hat ebenfalls eine spezielle Voraussetzung, hier sollte zuvor eine ATAM Analyse stattgefunden haben. Weitere Ausnahmen bilden die SAEM, SAAMER und SBAR, bei denen das Vorhandensein eines implementierten Systems von Nöten ist, um die Analyse mit den vorgesehenen Techniken durchzuführen.

Hilfsmittel sind nur bei wenigen Methoden angegeben. So liegen sie zum Beispiel in Form von Templates bei den Methoden QAW und FAAM vor. Bei der ATAM werden die Utility-trees als Hilfsmittel für die Analyse eingeführt, bei der ARID die Active Design Reviews, bei der AQA eine Scoring Tabelle und bei der APSM Berechnungsformeln. Die ASAAM bietet eine Menge von Regeln zur Unterstützung der Evaluation an und bei der CBAM werden zusätzlich zu den Prozessbeschreibungen auch Grundlagen zu wirtschaftlichen Sachverhalten angegeben, um die Evaluation zu erleichtern.

Zeitpunkt der Evaluation (Projektphase)

Beim Zeitpunkt der Durchführung einer Architekturbewertung gibt es wieder große Gemeinsamkeiten bei einem Großteil der Methoden. So ist in der Regel vorgesehen, die Evaluation zum Ende der Entwurfsphase der Architektur durchzuführen oder als iterativen Prozess während der Entwurfsphase umzusetzen (bspw. ATAM). Klare Ausnahmen sind hierbei die Methoden, die nicht streng auf Szenarien als Bewertungsansatz setzen oder sich mit den Attributen der Wiederverwendbarkeit eines Systems/einer Architektur beschäftigen. Aufgrund der Notwendigkeit für diese Methoden bereits ein implementiertes System zu haben, werden sie erst sehr spät im Entwicklungsprozess (SAEM) oder gar erst nach der Inbetriebnahme des entwickelten Systems durchgeführt (SBAR, SAAMER).

Benötigte Kenntnisse und Erfahrung des Evaluationsteams

Nur bei wenigen der zum Vergleich stehenden Bewertungsverfahren, sind Angaben oder zumindest Andeutungen zum nötigen Kenntnisstand des Evaluationsteams vorhanden. Die Kenntnisse über die jeweilige Qualitätsbewertungsmethode sind natürlich in der Beschreibung der Methode enthalten, zusätzlich werden aber dennoch bei einigen der Methoden bestimmte Kenntnisse vorausgesetzt. So werden zum Beispiel bei der ATAM Kenntnisse zum Aufstellen des Utility-tree benötigt, wohingegen bei der SAAM solche Fähigkeiten nicht benötigt werden, da die Methode an sich sehr einfach aufgebaut ist. Die ARID setzt voraus, dass die Verfahren der ATAM und ADR bekannt sind, da sie auf diesen basiert. Kenntnisse zu Messungen oder Metriken sind bei der AQA nötig. Bei der SACAM muss das Team in der Lage sein, wichtige

Qualitätsattribute durch Analyse der Geschäftsziele zu identifizieren, also ist auch ein Verständnis von wirtschaftlichen Zusammenhängen nötig. Besonders bei der SBAR Methode sind jedoch viele Kenntnisse nötig, da mehrere verschiedene Bewertungsansätze verwendet werden, die beherrscht werden müssen. Insgesamt bleibt festzuhalten, dass natürlich bei jeder Methode eine gewisse Grundkenntnis zur Qualität von Architekturen, Qualitätsattributen und informatischen Zusammenhängen vorausgesetzt werden muss.

Benötigte Ressourcen

Zu den benötigten Ressourcen für die Durchführung einer Architekturevaluation, lässt sich soweit überhaupt Angaben vorhanden sind, nur sagen das zumindest einige Tage für Meetings und Workshops mit Stakeholdern benötigt werden. Bei der ATAM wird beispielhaft von mindestens 3 Tagen gesprochen, der Zeitraum für andere Verfahren ist ähnlich. Es gibt jedoch auch Quellen die eine Zeitspanne von mehreren Wochen für ein Bewertungsverfahren ansetzen. Auch die Größe des Evaluationsteams wird nur sporadisch in den Methodenbeschreibungen erwähnt. Es ist anzunehmen, dass diese Ressourcen von verschiedenen Faktoren, wie Größe des Projektes, Umfang der Architektur, Menge der Stakeholder und Rolle des Evaluationsteams abhängen. Außerdem muss beachtet werden, dass die Vorbereitung einer Evaluation und die Nachbetrachtung (Reporting) durch das Evaluationsteam hier nicht mit eingerechnet sind. Eine allgemeine Aussage über Dauer oder Ressourcenbedarf eines Bewertungsverfahrens kann also nicht getroffen werden, da sie einfach zu variabel sind.

Ergebnisse (Output)

Die Ergebnisse, die von den jeweiligen Methoden produziert werden, richten sich einerseits nach den zuvor ausgegebenen Zielen und andererseits auch nach den angewendeten Bewertungsansätzen. So entsteht bei allen Szenario-basierten Methoden eine Sammlung von Szenarien unterschiedlicher Ausprägung, die an sich schon ein wichtiges Ergebnis des Verfahrens darstellt, da sie zum Verständnis der Architektur und der Qualitätsmerkmale wesentlich beitragen kann. Des Weiteren werden aus diesen Szenarien die angestrebten Aufwandsabschätzungen oder Auswirkungsbeschreibungen abgeleitet (APSM, ALMA, SAAMCS, SAAMER). Ein weiterer wichtiger Output einer Evaluation kann die Identifikation der Qualitätsmerkmale selbst sein (QAW, ASSAM). Und auch Messwerte sind bei einigen Methoden als Resultat vorzufinden (SAEM, PASA, AQA). Bei ein paar Methoden sind

Verbesserungsvorschläge (PASA, FAAM) oder Kostenanalysen (CBAM, PASA) als Teil des Ergebnisses vorgesehen. Die Identifikation von Risiken, Stärken und Schwächen einer Architektur stehen ebenfalls am Ende des Evaluationsprozesses (ATAM, SAE). Besonderheiten gibt es bei der Methode SBAR, die ein Reengineering der Architektur als Resultat vorsieht, sowie bei der SACAM, deren Ergebnis ein Vergleich mehrere Architekturen ist.

Anwendungsbereich (Domäne)

Bis auf zwei Ausnahmen sind bei allen Methoden keinerlei Einschränkungen im Bezug auf den Anwendungsbereich vorzufinden. Lediglich die SAE und die FAAM weichen hier ab. Die SAE wurde für die Evaluation von Telekommunikationssystemen konzipiert, während die FAAM mit dem Fokus auf Informationssysteme und -Systemfamilien entwickelt wurde.

Ausmaß der Evaluation

Nur bei der SACAM geht das Ausmaß eines Evaluationsverfahrens zwingend über die Analyse einer einzelnen Architektur hinaus. Es ist immer ein Vergleich mehrerer Architekturkandidaten vorgesehen, der in einem Durchlauf der Methode durchgeführt wird, wie sich auch in der Prozessbeschreibung widerspiegelt. Anderen Methoden beschränken sich auf eine Architektur pro Durchlauf (bspw. ATAM), was jedoch nicht heißt, dass mit ihnen nicht mehrere Architekturen verglichen werden können. Dazu müssen dann aber eben mehrere Durchläufe/Iterationen der Methode vorgenommen werden. Das Ausmaß der einzelnen Evaluation liegt hier aber immer nur bei einer Architektur. Und wieder andere Methoden können sowohl zur Analyse einzelner Architekturen oder zum Vergleich mehrerer Kandidaten genutzt werden (bspw. SAAM).

6 Ergebnisse

Im Laufe der vorherigen Kapitel wurden die nötigen Grundlagen gelegt, Qualitätsbewertungsmethoden verschiedener Art vorgestellt, Kriterien zum Vergleich dieser Methoden festgelegt und letztendlich der eigentliche Vergleich durchgeführt. Nun sollen zum Abschluss die Ergebnisse dieses Vergleichs zusammengetragen werden.

Zunächst ist festzuhalten, dass zwischen den Methoden durchaus Unterschiede bestehen. So sind die zu analysierenden Qualitätsmerkmale einer der größten Unterscheidungsfaktoren der Methoden. Je nach Fokus der Analyse können unterschiedlichste Aspekte einer Architektur untersucht werden, es gibt Methoden speziell für die Evaluation der Modifizierbarkeit oder des Instandhaltungsaufwands und es gibt Methoden, die zur Analyse diverser Qualitätsmerkmale auf einmal dienen. Dadurch sind auch in der Zielsetzung der Methoden Unterschiede im Detail festzustellen, die sich natürlich auch in den Ergebnissen der Methoden widerspiegeln.

Auch der Reifegrad der Methoden und ihre jeweilige Validierung stellen einen Unterschied dar. Zum einen gibt es ausgereifte Methoden wie die SAAM oder ATAM, die sowohl von ihrer Entwicklung her ausgereift sind, als auch in der Praxis bereits häufig erfolgreich angewendet wurden. Zum anderen sind viele der Qualitätsbewertungsmethoden kaum über die Konzeptions-/Entwicklungsphase hinaus entwickelt und Validierungen sind nur spärlich oder gar nicht vorhanden. Dieser Umstand ist auch bei der Untersuchung der Prozessbeschreibung klar festzustellen. Während einige Methoden ausführlich beschrieben werden, die Prozesse Schritt-für-Schritt erläutert sind und Fallstudien oder Beispiele vorhanden sind, gibt es zu anderen Methoden nur die grobe Beschreibung des vorgesehenen Konzepts.

Die Bewertungsansätze der Methoden sind zwar auch unterschiedlich ausgeprägt, doch als große Gemeinsamkeit lässt sich hier die Verwendung von Szenarien herausstellen. Fast alle Methoden nutzen, wenn auch nicht ausschließlich, Szenarien zur Architekturanalyse, einzig die AQA und SAEM verzichten gänzlich auf Szenarien. Ebenfalls bei nahezu allen Methoden vorhanden, ist die Forderung nach der Stakeholderbeteiligung, wenn auch in unterschiedlichem Ausmaß.

Beim Zeitpunkt für die Evaluation lassen sich zwei Klassen ausmachen, einerseits die frühen Methoden, die schon in der Entwurfsphase durchgeführt werden sollen und andererseits die späten Methoden, die erst sehr spät während der Entwicklung oder sogar erst während/nach der Anwendung des entwickelten Systems zum Einsatz kommen. Damit einhergehend unterscheiden sich auch die Voraussetzungen der zwei

Gruppen, es wird entweder ein Architekturentwurf benötigt (früh) oder eine (nahezu) komplette Implementierung (spät).

Zu den benötigten Ressourcen oder den nötigen Kenntnissen des Evaluationsteams lassen sich kaum Aussagen treffen, da zum einen einfach keine Angaben vorhanden sind, zum anderen aber auch die Variabilität der Methoden (was ihre Anwendung betrifft) keine genauen Angaben zulassen.

Außerdem ist noch erwähnenswert, dass fast alle Methoden keinerlei Einschränkung bezüglich ihres Anwendungsgebiets unterliegen, sie sind also bei jedem System/jeder Systementwicklung anwendbar. Viele der Methoden sind sowohl zur Analyse einzelner Architekturen als auch zum Vergleich mehrerer Architekturen nutzbar, lediglich eine der Methoden, die SACAM, erfordert es, immer mehrere Architekturkandidaten in einem Durchlauf miteinander zu vergleichen.

Es gibt also eine Vielzahl von unterschiedlichen Qualitätsbewertungsmethoden, die zur Bewertung einer Architektur genutzt werden können. Ihre Anwendung ist in der Regel kostengünstig, da zum Großteil Szenarien genutzt werden. Die Evaluation der Architektur kann großen Einfluss auf die Qualität des späteren Systems haben, da Risiken und Probleme aufgedeckt werden können. Alle vorgestellten Methoden würden zur Qualitätssicherung beitragen können, wenn auch einige eher ausgereift erscheinen als andere. So sind vor allem die SAAM und ATAM als verbreitete und inzwischen auch in Fachbüchern beschriebene Methoden hervorzuheben. Der erstellte Vergleich ermöglicht es, einen Überblick über die Qualitätsbewertungsmethoden zu erhalten, ausführlichere Informationen zu den jeweiligen Methoden sind in den jeweiligen Arbeiten, in denen sie vorgestellt werden, zu finden.

Literaturverzeichnis

- Andresen, A. (2004): Komponentenbasierte Softwareentwicklung mit MDA, UML 2 und XML, Hanser Verlag, München
- Babar, M./Zhu, L./Jeffrey, R. (2004): A Framework for Classifying and Comparing Software Architecture Evaluation Methods, In: Software Engineering Conference Proceedings, IEEE Computer Society Press 2004, S. 309-318
- Barbacci, M./Ellison, R./Lattanze, A./Stafford, J./Weinstock, C./Wood, W. (2003): Quality Attribute Workshops (QAW), Technical Report, SEI (CMU/SEI-2003-TR-016), Carnegie Mellon University, Pittsburgh
- Bass, L./Clements, P./Kazman, R. (2003): Software Architecture in Practice, Addison-Wesley, Boston, 2. Edition
- Bengtsson, P./Bosch, J. (1998): Scenario-based Software Architecture Reengineering, In: Proceedings of the 5th International Conference on Software Reuse, IEEE Computer Society Press 1998, S. 308-317
- Bengtsson, P./Bosch, J. (1999): 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
- Bengtsson, P./Lassing, N./Bosch, J./van Vliet, H. (2002): Architecture-level modifiability analysis (ALMA), In: The Journal of Systems and Software 69 (2004), S. 129-147
- Clements P., Kazman R., Klein M. (2002): Evaluating Software Architectures – Methods and Case Studies, Addison-Wesley, Boston
- Clements, P. (2000): Active Reviews for Intermediate Designs, Technical Note, SEI (CMU/SEI-2000-TN-009), Carnegie Mellon University, Pittsburgh
- Dobrica, L./Niemelä, E. (2002): A Survey on Software Architecture Analysis Methods, In: IEEE Transactions on Software Engineering, Vol. 28, No. 7, July 2002, S. 638-653
- Dolan T. (2001): Architecture Assessment of Information-System Families: a practical perspective, Proefschrift, Eindhoven University Press

- Dueñas, J./de Oliviera, W./ de la Puente, J. (1999): 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
- Eicker, S./Hegmanns, C./Malich, S. (2007): Auswahl von Bewertungsmethoden für Softwarearchitekturen, In: ICB-Research Report No. 14 (März 2007), Universität Duisburg-Essen
- Enzyklopädie der Wirtschaftsinformatik Webseite: Lexikon, abgerufen im Juli 2009, <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon>
- Hasselbring, W. (2006): Aktuelles Schlagwort: Software-Architektur, In: Informatik-Spektrum 29 (1), 2006, S. 48-52
- Heinrich, L./Heinzl, A./Roithmayr, F. (2004): Wirtschaftsinformatik-Lexikon, Oldenbourg Verlag, München, 7. Auflage
- Herden, S./Gomez, J./Rautenstrauch, C./Zwanziger, A. (2006): Software Architekturen für das E-Business, Springer Verlag, Berlin
- Ionita, M./Hammer, D./Obbink, H. (2002): Scenario-Based Software Architecture Evaluation Methods: An Overview, Research Paper, University Eindhoven
- Kazman, R./ Asundi, J./Klein, M. (2002): Making Architecture Design Decisions: An Economic Approach, Technical Report, SEI (CMU/SEI-2002-TR-035), Carnegie Mellon University, Pittsburgh
- Kazman, R./Klein, M./Clements, P. (2000): ATAM: Method for Architecture Evaluation, Technical Report, SEI (CMU/SEI-2000-TR-004), Carnegie Mellon University, Pittsburgh
- Lassing, N./Rijsenbrij, D./van Vliet, H. (1999): On Software Architecture Analysis of Flexibility, Complexity of Changes: Size isn't Everything, In: Proceedings of the 2nd Nordic Workshop on Software Architecture (1999), Ronneby (Sweden)
- Lung, C./Bot, S./Kalaichelvan, K./Kazman, R. (1997): An Approach to Software Architecture Analysis for Evolution and Reusability, In: Proceedings of CASCON (Centre for Advanced Studies Conference), Toronto (Canada), Oktober 1997, S. 144-154

- Molter, G. (1999): Integrating SAAM in Domain-centric and Reuse-based Development Processes, In: Proceedings of the 2nd Nordic Workshop on Software Architecture (1999), Ronneby (Sweden)
- SEI Website: Architectures, abgerufen im Juli 2009,
<http://www.sei.cmu.edu/architecture/>
- Starke, G. (2008): Effektive Software-Architekturen: Ein praktischer Leitfaden, Hanser Verlag, München
- Stoermer, C./Bachmann, F./Verhoef, C. (2003): SACAM: The Software Architecture Comparison Analysis Method, Technical Report, SEI (CMU/SEI-2003-TR-006), Carnegie Mellon University, Pittsburgh
- Tekinerdoğan, B. (2004): ASAAM: Aspectual Software Architecture Analysis Method, Research Paper, Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA'04), 2004
- Thiel, S. (2005): A Framework to Improve the Architecture Quality of Software-Intensive Systems, Dissertation, Universität Duisburg-Essen
- Vasconcelos, A./Sousa, P./Tribolet, J. (2005): Information System Architecture Evaluation: From Software to Enterprise Level Approaches, Im Rahmen der: 12th European Conference On Information Technology Evaluation (ECITE 2005), Finnland
- Williams, L./Smith, C. (2002): PASA: A Method for the Performance Assessment of Software Architectures, In: Proceedings of the Workshop on Software and Performance (WOSP2002), Rome (Italy), July 2002

Abschließende Erklärung

Ich versichere hiermit, dass ich die vorliegende Studienarbeit selbständig, ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Frank Eichler

Magdeburg, den 28. Juli 2009