



□ Ramon Anger

(ramon.anger@capgemini.com)

arbeitet als Technischer Architekt im Bereich Public der Capgemini Deutschland GmbH. Er beschäftigt sich insbesondere mit neuen Technologien, Softwarearchitektur und agilen Methoden.

## Vom Großen zum Ganzen – Architektur-Reviews mit ATAM & Co

Anwendungslandschaften mittlerer bis großer Unternehmen bestehen heute nicht selten aus hundert und mehr IT-Systemen, die sich in der Summe aus Milliarden Zeilen Quellcode zusammensetzen. Diese IT-Systeme basieren auf verschiedenen Technologien, stammen von unterschiedlichen Herstellern und erfüllen völlig unterschiedliche Zwecke. Auf dem Weg zur IT der Zukunft sind Unternehmen oft dazu angehalten, IT-Systeme miteinander zu vergleichen, z. B., um ihre vielförmige Anwendungslandschaft zu vereinheitlichen. Wichtige Voraussetzung für diese Gegenüberstellung ist, dass die IT-Systeme auch vergleichbar sind. Gegenwärtig erscheint es beispielsweise nicht sinnvoll, ein stark transaktionales Online-System auf funktionaler Ebene mit einem reinen Informationssystem zu vergleichen. Die Vergleichbarkeit ist hier eher auf der Ebene von Qualitätsmerkmalen wie Wartbarkeit, Benutzbarkeit oder Skalierbarkeit herstellbar. Dafür kommen Methoden zur Architekturbewertung wie die Architektur Tradeoff-Analyse (ATAM) & Co. ins Spiel, mit deren Hilfe IT-Systeme aneinander gemessen werden können. Dieser Artikel gibt einen kurzen Überblick über aktuelle Methoden zur Architekturbewertung und ihre Anwendbarkeit auf den Vergleich von IT-Systemen.

### Warum werden Architekturbewertungen durchgeführt?

Für die Durchführung einer Architekturbewertung eines IT-Systems gibt es verschiedene Gründe: die Prüfung der Tragfähigkeit eines Entwurfs, die Untersuchung konkreter bzw. gefühlter Architekturprobleme und die Vergleichbarkeit von Architekturentwürfen oder IT-Systemen.

Mit einer Architekturbewertung werden im Idealfall Risiken und Schwachstellen der Architektur eines Softwaresystems aufgedeckt. Für die Beseitigung oder Abschwächung dieser Risiken können dann geeignete Maßnahmen identifiziert und angewendet werden. Anschließend wird die Veränderung der Risiken mit geeigneten Mitteln gemessen. Ein Architektur-Review kann aber auch Informationen darüber liefern, wie gut Anforderungen an die Architektur eines Systems erfüllt werden (siehe unten, Abschnitt Eigenschaften, An-

forderungen und Rahmenbedingungen). Dieser Aspekt spielt beim Vergleich mehrerer Softwaresysteme ebenfalls eine Rolle.

### Auswahl der geeigneten Review-Methode

Im Internet sind aktuell etwa 100 Methoden zur Bewertung von IT-Architekturen dokumentiert. Eine sichtbare Verbreitung haben allerdings nur wenige dieser Methoden. Spitzenreiter bei der Verwendung sind aus unserer Sicht die Architecture Tradeoff Analysis Method (ATAM) und die Cost Benefit Analysis Method (CBAM).

Der Fokus der meisten Bewertungsansätze liegt auf einem einzelnen IT-System oder einem Teilsystem. Einige Bewertungsmethoden, wie die Software Architecture Comparison Analysis Method (SACAM) oder das Domain-Specific Software Architecture Comparison Model (DoSAM),

sind explizit für den Vergleich mehrerer Systeme konzipiert. Für diesen Zweck können allerdings die meisten der Bewertungsmethoden eingesetzt werden. [Tabelle 4](#) enthält einen kurzen Überblick über einige Architektur-Review-Methoden und ihre Eignung.

Der Leitspruch „Für jede Aufgabe das richtige Werkzeug“ gilt auch für die Auswahl der geeigneten Review-Methode. Die Auswahl ist von einer Vielzahl von Faktoren abhängig, z. B. welche Informationen über die Architektur der zu untersuchenden Systeme existieren, in welcher Form sie vorliegen und ob sie gleichermaßen über alle Subsysteme und Komponenten eines Systems in der notwendigen Granularität und Qualität zur Verfügung stehen. Diese Faktoren werden nicht von einem einzelnen System, sondern von fünf, fünfzig oder mehr zu untersuchenden Systemen beeinflusst und bestimmen, wie die

Kategorie	Methode	Anwendbarkeit	Detailgrad	Projektphase aus Architektursicht	Prüfgegenstand
Frage	Fragebogen	Generell	Grob	Früh	Artefakte und Prozesse
Frage	Checkliste	Domänen-spezifisch	Anpassbar	Mittel	Artefakte und Prozesse
Frage	Anwendungsfall	System-spezifisch	Mittel	Mittel	Artefakte
Messung	Metriken	Generell oder Domänen-spezifisch	Fein	Mittel	Artefakte
Messung	Prototyp, Simulation	Domänen-spezifisch	Anpassbar	Früh	Artefakte

Tab. 1: Best-Practice-Methoden für die Erhebung von Architektur-Informationen nach [Abo]

Informationserhebung zum Review über diese Systeme erfolgen muss. **Tabelle 1** zeigt typische Methoden für die Informationserhebung im Rahmen eines Architektur-Reviews.

Wenn ein Architekturgutachten für die gesamte Anwendungslandschaft eines Unternehmens durchgeführt werden soll, erscheint es sinnvoll, die Bewertung auf einige wenige Qualitätsmerkmale zu beschränken, um den Umfang der Bewertung überschaubar zu gestalten. Mit dieser Beschränkung nimmt allerdings auch die Aussagefähigkeit des Architekturgutachtens ab.

Wichtig erscheinen die Wiederverwendung von Teilen der Reviews mehrerer IT-Systeme zur Minimierung des Gesamtaufwands und die Sicherstellung der Vergleichbarkeit der Ergebnisse aller Auswertungen. Dazu müssen für die Evaluierung eines jeden Systems exakt die gleichen Maßstäbe angelegt werden, damit die Ergebnisse nicht verfälscht werden und damit die Vergleichbarkeit nicht gegeben ist.

Handelt es sich bei den zu evaluierenden Systemen um Eigen- bzw. Individualentwicklungen neueren Datums, bestehen je nach verwendeter Software-Entwicklungsmethodik gute Chancen auf angemessene Informationen. Anders kann das bei Kaufprodukten oder Altsystemen aussehen. Wir müssen uns beispielsweise darauf verlassen, dass ein Hersteller dem Kunden die Eigenschaften seines Produktes in ausreichendem Maß offenlegt: Existiert das Unternehmen des Herstellers überhaupt noch? Arbeiten die Architektur-Know-how-Träger eines Altsystems noch für das Unternehmen oder genießen sie bereits ihren verdienten Ruhestand?

### Eigenschaften, Anforderungen und Rahmenbedingungen

Welche Informationen sind denn eigentlich für die Bewertung der Architektur eines Systems relevant? Dies ist grundsätzlich von der Review-Methode abhängig, denn nicht alle haben denselben Fokus. Qualitätsmerkmale können auf den ersten Blick nicht für alle IT-Systeme der Anwendungslandschaft gleich sein, da die IT-Systeme zu verschieden sind, z. B. weil verschiedene funktionale Anforderungen gelten oder verschiedene Rahmenbedingungen vorliegen:

- *System I* ist ein Leistungsverfahren für die Berechnung und Zahlung von Krankenkassenbeiträgen,
- *System II* unterstützt die Abwicklung von Fortbildungsangeboten (Funktionalität),
- *System III* ist wochentags von 7 Uhr bis 19 Uhr verfügbar, während *System IV* jeden Tag 24 Stunden online sein muss (Rahmenbedingung).

IT-Systeme werden auch künftig unterschiedliche funktionale Anforderungen besitzen und unterschiedlichen Rahmenbedingungen unterliegen. Ein möglicher Ansatz, Vergleichbarkeit herzustellen, ist es, funktionale Anforderungen und Rahmenbedingungen auszublenden und die Vergleichbarkeit aus Sicht nichtfunktionaler Anforderungen zu untersuchen. Typische nichtfunktionale Anforderungen (NFA) von IT-Systemen sind:

- Benutzbarkeit
- Zuverlässigkeit
- Korrektheit
- Sicherheit

- Wartbarkeit
- Wirtschaftlichkeit
- Skalierbarkeit.

Diese nichtfunktionalen Anforderungen stellen gleichzeitig die Qualitätsmerkmale der Architektur von IT-Systemen dar (vgl. [Fir08]). Wenn eine Anwendung wartbar sein soll, muss die Architektur des IT-Systems Wartbarkeit unterstützen. Tut sie dies nicht oder nicht in ausreichendem Maß, kann davon ausgegangen werden, dass die Anwendung nur mit hohem Aufwand oder gar nicht wartbar ist.

Um zu differenzieren, ob Qualitätsmerkmale erfüllt werden, werden IT-Systeme mit den meisten Review-Methoden auf Basis ihrer Architektur bewertet. Eine Ausnahme hiervon bildet beispielsweise die Software Architecture Analysis Method (SAAM), deren Fokus unter anderem in einer Bewertung funktionaler Anforderungen liegt.

Einen interessanten Ansatz bietet die Methodik ATAM: Bei ihr werden unter anderem Entwurfsentscheidungen in Abhängigkeit von nichtfunktionalen Anforderungen betrachtet. Wenn eine oder mehrere Entwurfsentscheidungen die Erfüllung einer NFA behindern, kann die betroffene Systemarchitektur diese NFA nicht vollständig unterstützen. Leider lassen sich die Entwurfsentscheidungen, z. B. von Kaufsystemen von Kunden, selten ableiten, sodass die Anwendung von ATAM zum Architektur-Review auf Eigenentwicklungen beschränkt bleiben dürfte.

### Priorisierung von nichtfunktionalen Anforderungen

Die gleichzeitige Erfüllung mehrerer nicht-funktionaler Anforderungen einer Soft-

warearchitektur kann zueinander im Widerspruch stehen. So kann beispielsweise die Erfüllung der Anforderung *Sicherheit* indirekt die Erfüllung der Anforderungen *Einfachheit* oder *Testbarkeit* erschweren. Diese Abhängigkeiten zwischen nicht-funktionalen Anforderungen können in verschiedenen Systemen unterschiedlich aufgelöst werden. Für den Vergleich mehrerer Systeme ist es erforderlich, eine Priorisierung der einzelnen NFA über alle Systeme hinweg vorzunehmen.

**Scoring**

Aus dieser Priorisierung kann ein Scoring, d. h. ein gewichtetes Bewertungssystem, abgeleitet werden. Das Scoring belegt, in welchem Maß zu untersuchende Systeme nichtfunktionale Anforderungen entsprechend der vorher festgelegten Priorisierung erfüllen. Die vollständige Erfüllung einer hoch priorisierten NFA verschafft einem System z. B. den Score 21, eine voll erfüllte niedrig priorisierte NFA den Score drei. Wenn eine Systemarchitektur eine NFA vollumfänglich erfüllt, erhält das System für die NFA die volle Punktzahl (21), bei mittlerem Erfüllungsgrad den Score zwei Drittel der vollen Punktzahl (14) und bei geringer Erfüllung ein Drittel der Maximalpunktzahl (7). Wenn ein System eine NFA gar nicht unterstützt, werden für diese NFA keine Punkte vergeben.

Grundlage für das Scoring ist die Festlegung, unter welchen Bedingungen eine NFA welchen Erfüllungsgrad für ein System hat. Eine Möglichkeit sind prozentuale Wertebereiche (siehe Tabelle 2). Hierfür bietet sich die Nutzung eines Kriterienkatalogs für jede NFA an. Die Kriterien müssen so präzise messbar sein, dass widerspruchsfrei entschieden werden kann, ob ein Kriterium erfüllt ist oder nicht. Vier von fünf möglichen erfüllten Kriterien entsprechen einem Erfüllungsgrad von 80 % für die betroffene NFA. Allerdings kennen die meisten Architektur-Review-Methoden keinen prozentualen Erfüllungsgrad einer NFA, sodass dieser Aspekt für einen vergleichenden Review von IT-Systemen ergänzt werden muss.

Die Evaluierung der Kriterien pro zu untersuchendem System kann je nach Form der für die Prüfung erforderlichen Informationen unter Nutzung verschiedenster Hilfsmittel erfolgen (siehe Tabelle 1). Tabelle 3 zeigt beispielhaft eine Bewertung für vier Systeme unter Benutzung des Bewertungsschemas aus Tabelle 2.

		Erfüllungsgrad			
		100 %-85 %	84 %-50 %	49 %-25 %	<24 %
Priorität	NFA	Hoch	Mittel	Gering	Nicht
1	Wartbarkeit	21	14	7	0
2	Benutzbarkeit	18	12	6	0
3	Betriebbarkeit	15	10	5	0
4	Performance	12	8	4	0
5	Testbarkeit	9	6	3	0
6	Stabilität	6	4	2	0
7	Portierbarkeit	3	2	1	0
Score max.		84			

Tab. 2: Scoring-Schema-Beispiel für priorisierte nichtfunktionale Anforderungen

NFA	max. Score	System I		System II		System III		System IV	
Wartbarkeit	21	Mittel	14	Mittel	14	Gering	7	Hoch	21
Benutzbarkeit	18	Hoch	18	Mittel	12	Hoch	18	Mittel	12
Betriebbarkeit	15	Gering	5	Mittel	10	Mittel	10	Gering	5
Performance	12	Mittel	8	Hoch	12	Gering	4	Gering	4
Testbarkeit	9	Hoch	9	Mittel	6	Gering	3	Mittel	6
Stabilität	6	Mittel	4	Gering	2	Mittel	4	Mittel	4
Portierbarkeit	3	Mittel	2	Hoch	3	Nicht	0	Mittel	2
Score	84		60		59		46		54

Tab. 3: Beispielbewertung für vier untersuchte Systeme I bis IV

Die Bewertung der Systeme ist völlig losgelöst von ihrer Funktionalität oder vom Technologie-Stack erfolgt. Welche Aussage kann aus der Beispielbewertung in Tabelle 3 abgeleitet werden?

In erster Linie, dass die untersuchten Systeme die priorisierten nichtfunktionalen Anforderungen unterschiedlich gut unterstützen. Wartbarkeit wird von System III nur in geringem Maß unterstützt. Wenn die Architekturbewertung hierfür noch keine Ursachen ans Licht gebracht hat, muss hier Ursachenforschung betrieben werden. Die Betriebbarkeit, obwohl mit Priorität drei versehen, wird von keinem der untersuchten Systeme vollumfänglich erfüllt. In dieser Situation gilt, es zu untersuchen, ob die Priorisierung dieser NFA richtig erfolgt ist.

**Fazit**

Viele Architektur-Review-Methoden legen ihren Fokus auf ein einzelnes IT-System oder Teilsystem. Trotzdem sind die meisten Methoden für den Vergleich mehrerer Systeme geeignet. Beim Vergleich ist es erforderlich, exakt dieselben Bewertungskriterien an alle untersuchten IT-Systeme anzulegen. Bei der Bewertung unter Nutzung von nichtfunktionalen Anforderungen sollten diese untereinander priorisiert sein sowie ein Scoring-Schema festgelegt werden. Dadurch wird NFA-Erfüllung zwischen den untersuchten Systemen auf Ebene der Architektur vergleichbar gestaltet. Um den Aufwand des vergleichenden Reviews überschaubar zu halten, wird empfohlen, nur die wesentlichen nichtfunktionalen Anforderungen zu untersuchen. ■

**Quellen**

**[Fir08]** Donald G. Firesmith et al: The Method Framework for Engineering System Architectures, Auerbach Publications, 2008  
**[Abo]** Abowd, G., L. Bass, P. Clements, R. Kazman, L. Northrop and A. Zaremski: Recommended Best Industrial Practice for Software Architecture Evaluation, <http://www.sei.cmu.edu/reports/96tr025.pdf>

Methode	Fokus	Eignung für Vergleich	Weiterführende Informationen
Software Architecture Analysis Method (SAAM)	Fokus liegt auf Abschätzung der Wartbarkeit und auf funktionalen Anforderungen. Beschränkung der betrachteten NFAs auf Sicherheit, Modifizierbarkeit, Portierbarkeit.	Besonders geeignet bei Systemen mit vergleichbarer Funktionalität, weniger für einen Vergleich vieler nicht-funktionaler Anforderungen.	<a href="http://www.sei.cmu.edu/library/assets/ICSE16.pdf">http://www.sei.cmu.edu/library/assets/ICSE16.pdf</a>
Architecture Tradeoff Analysis Method (ATAM)	Hat die Bewertung nichtfunktionaler Eigenschaften der Architektur als Gegenstand und unterstützt bei der Identifikation von Architekturrisiken.	Für eine Bewertung mehrerer Systeme auf Basis von nichtfunktionalen Anforderungen gut geeignet.	<a href="http://www.sei.cmu.edu/architecture/tools/evaluate/atam.cfm">http://www.sei.cmu.edu/architecture/tools/evaluate/atam.cfm</a>
Cost Benefit Analysis Method (CBAM)	Hat die Gegenüberstellung von Kosten, Nutzen und Risiken der Architektur zum Fokus. Baut auf den Ergebnissen von ATAM auf.	Besonders geeignet, wenn Kosten, Nutzen und Risiken für einen Vergleich mehrerer Systeme relevant sind.	<a href="http://www.sei.cmu.edu/architecture/tools/evaluate/cbam.cfm">http://www.sei.cmu.edu/architecture/tools/evaluate/cbam.cfm</a>
Quality Assessment of Software-Intensive System Architecture (QUASAR)	Grundannahme der Methode ist, dass sämtliche Qualitätsmerkmale eines IT-Systems auch für seine Teilsysteme und Teil-Teilsysteme überprüfbar sein müssen.	Systeme können gedanklich auf die Ebene von Teilsystemen verschoben werden und so die Erfüllung von nicht-funktionalen Anforderungen für die gesamte Anwendungslandschaft im Kontext eines Systems geprüft werden. Auf diese Weise ist die Methode ähnlich gut auf den Vergleich mehrerer Systeme anwendbar wie SACAM.	<a href="http://www.sei.cmu.edu/reports/06hb001.pdf">http://www.sei.cmu.edu/reports/06hb001.pdf</a>
Family Architecture Assessment Method (FAAM)	Zielt auf die Prüfung der Architektur einer Produktfamilie ab.	Für den Vergleich z. B. zwischen Produktfamilien geeignet.	<a href="http://alexandria.tue.nl/extra2/200111741.pdf">http://alexandria.tue.nl/extra2/200111741.pdf</a>
Software Architecture Comparison Analysis Method (SACAM)	Zielt auf den Vergleich mehrere Architektur-Alternativen für ein System vor dessen Implementierung ab. Versieht eine Alternative mit einem Score, der bestimmt, wie gut nichtfunktionale Anforderungen unterstützt werden.	Für eine Bewertung mehrerer Systeme nur auf Basis von nichtfunktionalen Anforderungen gut geeignet.	<a href="http://www.sei.cmu.edu/reports/03tr006.pdf">http://www.sei.cmu.edu/reports/03tr006.pdf</a>
Domain-Specific Software Architecture Comparison Model (DoSAM)	Fokus ist die Bereitstellung eines Domain-spezifischen Frameworks als Referenz für den Vergleich mit System-Architekturen innerhalb dieser fachlichen/technischen Domäne auf Basis von nicht-funktionalen Anforderungen.	Unter der Annahme, dass die Domäne und die resultierende Referenzarchitektur beliebig weit gefasst werden kann, prinzipiell universell einsetzbar. Allerdings besitzt eine universell einsetzbare Referenzarchitektur wenig Aussagekraft und die resultierende Bewertung ist nur oberflächlich möglich.	<a href="http://sse-world.de/index.php/download_file/view_inline/133/">http://sse-world.de/index.php/download_file/view_inline/133/</a>

Tab. 4: Übersicht über ausgewählte Methoden zum Architektur-Review mehrerer Systeme