



Fraunhofer Institut
Software- und
Systemtechnik

Architekturbewertung

Martin Große-Rhode,
Ekkart Kleinod, Stefan Mann



ISST-Bericht 82/07
Juli 2007

Herausgeber: Fraunhofer-Gesellschaft e. V.
Institut für Software- und Systemtechnik
Leitung: Prof. Dr. Jakob Rehof
Institutsteil Berlin: Mollstraße 1
10178 Berlin
Institutsteil Dortmund: Joseph-von-Fraunhofer-Straße 20
44227 Dortmund

ISSN 0943-1624



Das Projekt VEIA

Das Projekt »Verteilte Entwicklung und Integration von Automotive-Produktlinien« wird vom Bundesministerium für Bildung und Forschung im Rahmen der Forschungsoffensive »Software Engineering 2006« unter dem Förderkennzeichen »01ISF15A« gefördert.

Am Projekt sind vier Partner beteiligt:

- BMW Group
- Fraunhofer ISST
- PROSTEP IMP GmbH
- Technische Universität München

Webseite: <http://veia.isst.fraunhofer.de/>

Autoren

Dieses Dokument wurde geschrieben von:

Martin Große-Rhode, Ekkart Kleinod, Stefan Mann (Fraunhofer ISST).

Inhalt

1	Einleitung	3
1.1	Einordnung der Architekturbewertung in VEIA	3
1.2	Aufbau des Dokuments	3
2	Der VEIA-Referenzprozess	4
3	Grundlagen	6
3.1	Begriffsdefinition	6
3.2	Allgemeine Überlegungen	8
3.3	Informationsgehalt und Nutzen von Architekturmodellen	13
3.4	Vorgehen	14
4	Szenarios und Ziele	16
4.1	Beispielfragen	17
4.2	Beispiel: Produktlinie oder Produkt für CBS	18
5	Kriterien und Bewertungen	20
5.1	Beispielkriterien und deren Bewertungen	21
5.2	Beispiel: Produktlinie oder Produkt für CBS	21
6	Messgrößen und Messwerte	23
6.1	Beispielmessgrößen	23
6.2	Beispiel: Produktlinie oder Produkt für CBS	23
7	Metriken	24
7.1	Beispielmetriken	25
7.2	Beispiel: Produktlinie oder Produkt für CBS	26
8	Einbettung in den Entwicklungsprozess	27
Literatur		29

1 Einleitung

1.1 Einordnung der Architekturbewertung in VEIA

Das Projekt VEIA hat sich das Ziel gesetzt, auf der Grundlage der Konzepte der Produktlinientechnik eine Methode für die verteilte Entwicklung und Integration von Automotive-Systemen zu erarbeiten, die sich an den konkreten Anforderungen industrieller Entwicklungsprozesse orientiert und praktisch anwendbar ist.

In der ersten Projektphase wurde der Grobentwurf eines Referenzprozesses definiert, in dem unter anderem die Artefakte zur Erfassung und Modellierung von Architektursichten festgelegt sind. (Siehe [[GEKM07](#)].) Die Architekturmodelle sind dabei nicht Eins zu Eins den Phasen des Referenzprozesses zugeordnet, sondern werden kontinuierlich weiter entwickelt.

In der zweiten Phase des Projekts stand die Frage im Vordergrund, welchen Entwicklungsstand die Architekturmodelle zu bestimmten Zeitpunkten beziehungsweise zum Ende der Prozessphasen haben müssen. Dazu wurde untersucht, welche Informationen in den Architekturmodellen erfasst sein müssen, um bestimmte Eigenschaften des zu entwickelnden Systems bewerten zu können.

Dieses Dokument beschreibt die Ergebnisse der zweiten Projektphase.

1.2 Aufbau des Dokuments

Im folgenden [Kapitel 2](#) wird der Referenzprozess kurz rekapituliert. Danach, in [Kapitel 3](#) werden die Grundlagen der Architekturbewertung gelegt. Nach einer Begriffsbestimmung werden die Facetten der Architekturbewertung untersucht und dargestellt.

[Abschnitt 3.4](#) definiert ein Vorgehen für die Architekturbewertung. Aus diesem Vorgehen und den Grundlagen heraus werden die einzelnen Elemente der Architekturbewertung gewonnen, die in den nachfolgenden Kapiteln [4](#) bis [7](#) detailliert beschrieben werden.

2 Der VEIA-Referenzprozess

Dieses Kapitel stellt kurz den VEIA-Referenzprozess ([GEKM07]) vor.

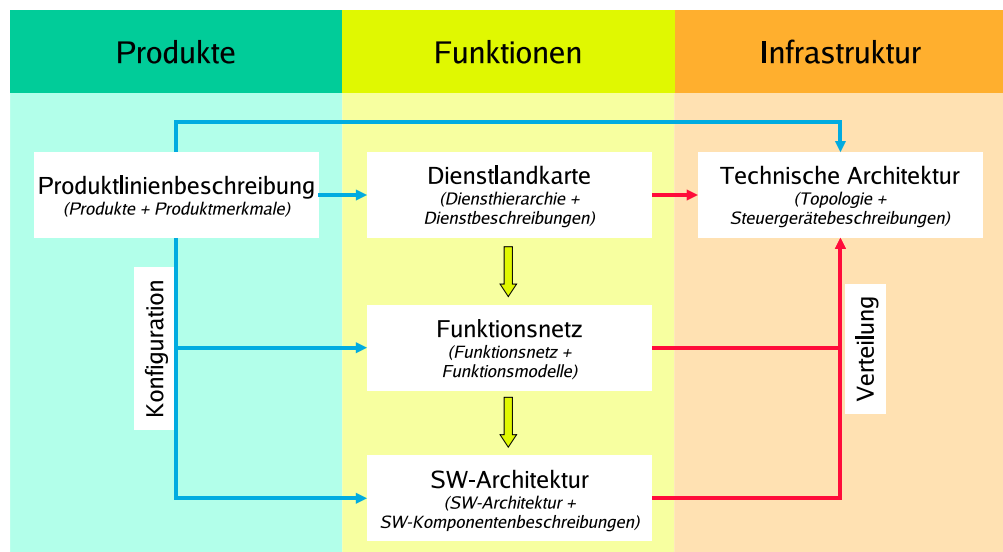
Abbildung 1 zeigt die Artefakte des VEIA-Referenzprozesses. Sie sind in folgende Bereiche unterteilt:

Produkte Beschreibung der Merkmale der zu entwickelnden Systeme,

Funktionen Beschreibung der zu implementierenden Funktionalität der Systeme,

Infrastruktur Beschreibung der technischen Ressourcen, die zur Umsetzung der Funktionalität verwendet werden.

Abbildung 1 Artefakte und ihre Beziehungen im VEIA-Referenzprozess.



Folgende Artefakte werden im Prozess erstellt:

Produktlinienbeschreibung definiert die Produktlinie, die entwickelt werden soll, indem die Features (Merkmale) der in der Produktlinie enthaltenen Produkte angegeben werden. Jedes Produkt innerhalb der Produktlinie ist durch eine zulässige Auswahl aus diesen Features definiert.

Dienstlandkarte definiert die Dienste, die die Systeme anbieten. Ein Dienst kann als ein Use-Case verstanden werden, der von einem System ausgeführt werden

kann. Die Dienstlandkarte enthält, den Diensten zugeordnet, die Anforderungen an die Systeme.

Funktionsnetz definiert die logische Architektur der Systeme, d. h. die Zusammensetzung der Dienste aus Teilfunktionen. Das Funktionsnetz zeigt den Informationsfluss und die logischen Beziehungen zwischen den einzelnen Funktionen.

Softwarearchitektur definiert die Softwarekomponenten, die den Softwareanteil der Systeme ausmachen, sowie deren Beziehungen. Die Softwarekomponenten der Softwarearchitektur realisieren die Funktionen des Funktionsnetzes.

Technische Architektur definiert die Hardwarekomponenten, die den Hardwareanteil der Systeme ausmachen. Dazu zählen insbesondere Steuergeräte, die gemäß einer Hardwaretopologie verbaut werden, sowie deren Kommunikationsverbindungen.

Konfigurationsmodelle bilden die Varianz des Produktlinienmodells auf die anderen Modelle ab. Damit können die Features den Komponenten der anderen Modelle zugeordnet sowie die Varianzbeschreibung im Produktlinienmodell für die Konfiguration der Dienst- und Architekturmodelle genutzt werden.

Verteilungsmodelle bilden die Modelle des Funktionsbereichs auf die technische Architektur ab. Damit wird das Deployment der einzelnen Komponenten auf die Hardware definiert.

Gegenstand der Architekturbewertung sind die Architekturmodelle: Dienstlandkarte, Funktionsnetz, Softwarearchitektur und technische Architektur. Die Produktlinienbeschreibung ist lediglich eine Featurebeschreibung. Sie enthält die Features des modellierten Systems sowie die kundensichtbare Varianz, stellt aber keine Sicht auf die Architektur des Systems dar. Die Konfigurationsmodelle, die die Beziehungen zwischen Features und Komponenten herstellen, sind dementsprechend ebenfalls keine Architekturmodelle.

Daher sind im Sinne der Architekturbewertung im Folgenden die Dienstlandkarte, das Funktionsnetz, die Softwarearchitektur, die technische Architektur sowie die Verteilungsmodelle, die die Beziehungen zwischen den verschiedenen Architekturen beschreiben, Gegenstand des Interesses.

3 Grundlagen

Bevor die Architekturbewertung diskutiert wird, sollen in diesem einführenden Abschnitt die Grundlagen der Architekturbewertung beschrieben werden, wie sie in diesem Dokument verwendet und verstanden werden. Den Anfang bildet eine kurze Begriffsdefinition, damit keine sprachlichen Missverständnisse auftreten. Der zweite Teil der Grundlagen sind allgemeine Überlegungen zur Architekturbewertung und den damit zusammenhängenden Fragen. Nach Überlegungen zu Information und Nutzen von Architekturmodellen für die Architekturbewertung bildet die Definition des Vorgehens bei der Architekturbewertung den Abschluss des Kapitels.

3.1 Begriffsdefinition

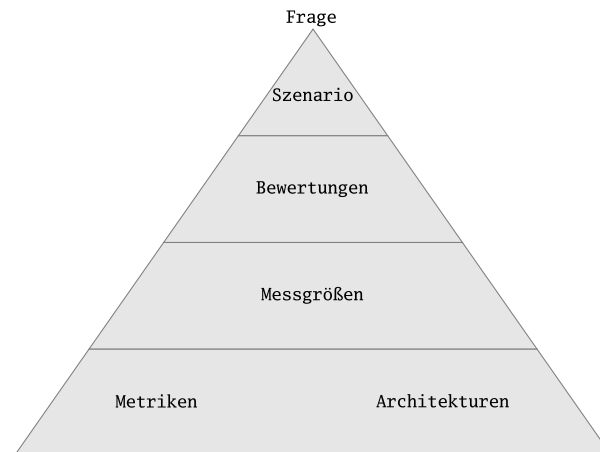
An dieser Stelle werden die im Dokument verwendeten Begriffe definiert, um Zweideutigkeiten zu vermeiden und eine eindeutige Sprachregelung innerhalb des Dokuments zu schaffen und zu motivieren.

Architekturen und Modelle

Architektur Eine Architektur eines E/E-Systems ist, allgemein gesprochen, die Beschreibung der Bestandteile des Systems sowie der Beziehungen zwischen den Bestandteilen. Welche Bestandteile und Beziehungen beschrieben werden, hängt vom Standpunkt des Betrachters sowie dem Ziel der Betrachtung ab. Das heißt, ein E/E-System kann, je nach Betrachtung, viele Architekturen haben. In VEIA unterscheiden wir die *Informationsarchitektur* (Dienstlandkarte und Funktionsnetz), die *Softwarearchitektur* sowie die *Hardwarearchitektur*. Weitere, in VEIA nicht betrachtete Architekturen von E/E-Systemen sind beispielsweise die Geometrie oder die Klassenstruktur der Software.

Architekturmodell Ein Architekturmodell beschreibt eine Architektur, es ist das Modell eines Systems unter einem bestimmten Blickwinkel. In VEIA wird für alle Architekturen vorgegeben, mit welchen Sprachmitteln die jeweilige Architektur zu beschreiben ist. Oft wird auch der Begriff Architekturbeschreibung statt Architekturmodell verwendet.

Abbildung 2 Elemente der Architekturbewertung.



Szenarios, Bewertungen und Metriken

Im Folgenden wird die Architekturbewertung analysiert und in Teile zerlegt, die in [Abbildung 2](#) gezeigt werden. Die einzelnen Elemente werden in den Kapiteln [4](#) bis [7](#) ausführlich erläutert, zum Verständnis sei nachfolgend eine kurze, nicht vollständige Erläuterung gegeben.

Szenario/Ziel »Welche Frage soll beantwortet werden?«

Das Ziel der Architekturbewertung wird durch eine Zielfrage ausgedrückt, deren Beantwortung Aufgabe der Bewertung ist. Zu jeder Architekturbewertung wird ein Szenario eröffnet, das alle Artefakte der Architekturbewertung zusammenhält. Im Szenario werden die notwendigen Kriterien ermittelt, die mit Hilfe der einzelnen Bewertungen beantwortet werden müssen.

Bewertung »Was bedeuten die Messergebnisse?«

Die Bewertung selbst ist die Interpretation der Messergebnisse, die für die Bewertung ermittelt wurden. Die Messwerte werden eingeschätzt, in Relation gebracht – bewertet. Für jede Bewertung sind die Messgrößen anzugeben, deren Werte für die Bewertung benötigt werden.

Messgröße, Messwert »Was wird gemessen?«

Eine Messgröße ist eine Eigenschaft einer Architektur. Der Wert der Eigenschaft wird durch den Messwert der Messgröße ausgedrückt. Zur Ermittlung der Messgrößen sind Metriken erforderlich, die für jede Architektur bzw. jedes Architekturmodell festlegen, wie der Messwert gemessen wird.

Metrik »Wie wird gemessen?«

Eine Metrik ist eine Funktion, die aus gegebenen Parametern einer oder mehrerer zugrundeliegender Architekturmodelle den Wert einer *Messgröße* ermittelt. Eine Metrik wird meist spezifisch für eine Art von Architekturmodellen aufgestellt und ist auch nur für diese gültig. Ein Messwert kann mit einer Metrik gemessen oder geschätzt (prognostiziert, hochgerechnet) werden. Die Güte einer Metrik ist für die Bewertung der Messwerte wichtig.

3.2 Allgemeine Überlegungen

Nachdem die verwendeten Begriffe definiert und erläutert wurden, soll an dieser Stelle die Bandbreite der relevanten Themen bezüglich der Architekturbewertung gezeigt werden.

Bei einer Bewertung wird der zu bewertende Gegenstand anhand von Kriterien untersucht und bezüglich einer bestimmten Skala oder bezüglich einer Referenz eingeschätzt. Übertragen auf Architekturen bedeutet das, Architekturbewertung ist die Einschätzung des Werts oder der Qualität einer Architektur.

Im Rahmen des VEIA-Referenzprozesses werden Architekturbewertungen zu unterschiedlichen Zwecken durchgeführt. Neben der Bewertung einzelner Architekturentwürfe umfasst die Architekturbewertung sowohl den Vergleich mehrerer Architekturentwürfe als auch den Vergleich mehrerer Realisierungsentwürfe für eine Produktlinie. Eine weitere Unterscheidung betrifft den Gegenstand der Bewertung. Architekturbewertungen können Aussagen über modellierte Systeme treffen, sie können aber auch Aussagen über die Qualität der Modellierung, also die Modelle an sich liefern. Drittens spielt der Zeitpunkt der Bewertung eine Rolle. Je nachdem, ob eine Bewertung früh im Prozess auf unvollständig spezifizierten Architekturen oder spät auf nahezu vollständigen Architekturmodellen durchgeführt werden soll, sind andere Aussagen möglich, andere Metriken zu verwenden etc.

Wir unterscheiden also drei grundlegende Dimensionen der Architekturbewertung:

- 1 Zweck,
- 2 Gegenstand und
- 3 Zeitpunkt der Bewertung.

3.2.1 Zweck der Bewertung

Der Zweck der Architekturbewertung bestimmt, welche Art der Bewertung vorgenommen wird. Er wird oft durch eine Frage ausgedrückt, deren Beantwortung das Bewertungsszenario beschließt.

Zunächst gibt es Fragen, die genau ein Architekturmodell bzw. genau ein System betreffen. Eine solche Frage wäre z. B. »Welche Kosten würde mein modelliertes System in der Implementierung verursachen?« Zur Beantwortung dieser Fragen werden die entsprechenden Messgrößen für die Architektur bestimmt, gemessen und bewertet. Eine eventuelle Bewertung wird anhand eines Referenzwerts vorgenommen. Im Sinne von Aufgaben–Lösungen betrachtet: Über Anforderungen wird eine Aufgabe an die Architektur gestellt, die von der Lösung, dem Architekturentwurf erfüllt werden muss.

Zweitens gibt es Fragen, die zwei oder mehrere Architekturentwürfe bzw. Systeme betreffen. Sie sind meist Vergleichsfragen, die die Architekturen unter vorgegebenen Gesichtspunkten bewerten wollen, um daraus einen Vergleich ziehen zu können. Ein Beispiel für solch eine Frage wäre: »Welches der beiden modellierten Produkte ist kostengünstiger herzustellen?« Die Beantwortung dieser Fragen wird zunächst auf die Beantwortung der Einzelfragen für die jeweilige Architektur zurückgeführt: »Wieviel kostet Produkt 1?« und »Wieviel kostet Produkt 2?« Danach werden die ermittelten Werte verglichen und so die Antwort auf die Ursprungsfrage ermittelt. Es werden also zwei Architekturentwürfe miteinander verglichen.

Die dritte Art von Zielen innerhalb des VEIA-Referenzprozesses betrifft die Bewertung von Softwarearchitekturen für Produktlinien. Diese Bewertungen zielen nicht mehr auf alle Arten von Architekturmodellen ab, wie die Einzel- und Vergleichsbewertung oben, sondern auf die Umsetzung einer Funktionsarchitektur (eines Funktionsnetzes) in eine Softwarearchitektur.

Die Fragen, die hier beantwortet werden sollen, sind Fragen, die eine optimale oder gute Lösung unter verschiedenen Lösungsmöglichkeiten suchen. Die in Frage kommenden Lösungsalternativen sind dabei nicht vorgegeben, sondern müssen aus der Menge aller möglichen Realisierungsentwürfe für die Produktlinie ermittelt werden. Eine Beispielfrage wäre: »Ist es günstiger, für Condition Based Service (CBS) eine Software-Produktlinie mit für jedes Fahrzeug angepassten Softwarekomponenten zu implementieren oder universelle Softwarekomponenten, die in jedem Fahrzeug parametrisiert werden müssen? Wenn die Lösung eine Software-Produktlinie ist: Wie wird die fahrzeugspezifische Varianz in der Softwarearchitektur und den Softwarekomponenten umgesetzt?«

Zur Beantwortung dieser Frage muss zunächst ermittelt werden, welche Softwareentwürfe miteinander verglichen werden sollen. Das heißt, es muss der Lösungsraum der Architektur aufgespannt werden. Dabei ist zu beachten, dass bereits bei der Bestimmung des Lösungsraums eine Vorauswahl stattfindet: es sollen nur sinnvolle Lösungen in den Lösungsraum übernommen werden. Werden alle möglichen Lösungen übernommen und bewertet, ist bei der Vielzahl an Varianten eines modellierten Systems wegen der stark steigenden Zahl von Lösungen kein Ende der Bewertung abzusehen. Die Vorauswahl der Lösungen, die betrachtet werden sollen, muss vor der Bewertung getroffen werden und kann als Designschritt verstanden werden. Im Allgemeinen ist der Lösungsraum der Architektur nicht von der Bewertungsfrage abhängig, sondern für die Produktlinie festgelegt. Nach der Definition des Lösungsraums wird wie bereits bekannt vorgegangen: Die Einzelfrage wird für jede Lösung beantwortet, die Ergebnisse verglichen und so die Antwort auf die Ursprungsfrage ermittelt.

Es wird also eine Aufgabe gestellt, für deren Lösung es mehrere Architekturentwürfe gibt. Diese Architekturentwürfe werden einzeln gemessen und im Hinblick darauf bewertet, welcher Entwurf die beste Lösung ist.

Ein Beispiel zur Verdeutlichung: Sei unsere Softwareproduktlinie eine Motorsteuerung, deren Varianten Otto-, Diesel- und Hybridmotoren ansteuern können. Dann gibt es fünf Strukturierungsmöglichkeiten der Softwarekomponenten:

- 1 eine Komponente für alle Steuerungen (die 150 %-Lösung)
- 2 eine Komponente für Otto-Motoren, eine für Diesel- und Hybridmotoren
- 3 eine Komponente für Dieselmotoren, eine für Otto- und Hybrid-Motoren
- 4 eine Komponente für Hybridmotoren, eine für Otto- und Diesel-Motoren
- 5 eine Komponente für Otto-Motoren, eine für Diesel-Motoren und eine für Hybridmotoren

Dieser maximale Lösungsraum wird nun durch den Systementwickler auf die Lösungen eingeschränkt, die betrachtet werden sollen. Welche Lösungen das sind, ist wieder eine Bewertungsaufgabe, die aufgrund verschiedenster Kriterien beantwortet werden kann: Machbarkeit, Hardwarezulieferer, Ähnlichkeit der Implementierung, um nur einige zu nennen. Im Beispiel wird der Lösungsraum auf folgende Realisierungen eingeschränkt:

- 1 eine Komponente für alle Steuerungen (die 150 %-Lösung)
- 4 eine Komponente für Hybridmotoren, eine für Otto- und Diesel-Motoren
- 5 eine Komponente für Otto-Motoren, eine für Diesel-Motoren und eine für Hybridmotoren

Jetzt werden für jede Realisierung die Kosten ermittelt, die dann miteinander

verglichen werden, um so zu entscheiden, ob eine Produktlinie oder für jedes Fahrzeug angepasste Softwarekomponenten gebaut werden sollen. Die Ermittlung der Kosten für die Lösungen 4 und 5 ist etwas komplizierter, da eine Metrik die Bestandteile der Lösung einzeln messen und zu einem Gesamtwert zusammenführen muss, dieses Problem wird im Kapitel über Metriken ([Kapitel 7](#)) ausführlicher behandelt.

3.2.2 Gegenstand der Bewertung

Wenn von »Architekturbewertung« gesprochen wird, ist meist die Bewertung des mittels der Architektur modellierten Systems gemeint. Die Fragen, die dabei beantwortet werden, zielen auf Eigenschaften des Systems ab, wie Ausführungszeiten, Kosten oder Speicherbedarf. Wird eine Produktlinie modelliert, können zusätzliche Eigenschaften wie Grad der Varianz oder Ähnlichkeit bewertet werden. Des Weiteren gibt es Prozesseigenschaften wie Entwurfsaufwand oder Testaufwand, die ebenfalls bewertet werden müssen. Andererseits können auch Modelleigenschaften bewertet werden, wie Vollständigkeit, Reifegrad oder Konsistenz der Modellierung.

Je nach Gegenstand der Bewertung ändern sich die Informationen, die zur Beantwortung der Fragen herangezogen werden. Für die Architekturbewertung für ein System, eine Produktfamilie oder einen Prozess sind die Informationen heranzuziehen, die mit Hilfe der Architektur modelliert werden. Es sind also die in das Modell eingetragenen Informationen zu bewerten. Fehlende Informationen müssen über Schätzungen und Erfahrungswerte vorangegangener Modelle ersetzt werden. Die Bewertungen lassen also sowohl Messungen als auch Schätzungen zu und sind somit geeignet, auch zukünftige Eigenschaften eines Systems zur Bewertung heranzuziehen, vorausgesetzt natürlich, dass die Schätzungen gut genug sind.

Bei der Bewertung der Modelleigenschaften wird bewertet, *wie* modelliert wurde, nicht *was* modelliert wurde. Das heißt, der Reifegrad oder die Qualität des Modells werden bewertet, zum Beispiel die Vollständigkeit des Modells oder die Einhaltung von Modellierungsrichtlinien. Diese Informationen lassen sich schwer schätzen, da Modelle Momentaufnahmen der Modellierung sind. Bewertungen der Modelle lassen also hauptsächlich momentane Bewertungen von Entwicklungsständen zu, keine Schätzungen über zukünftige Modelle.

Mit Hilfe der Bewertung von Systemen, Produktfamilien oder Prozessen werden also Aussagen über zukünftige Eigenschaften des Systems getroffen, die durch

Schätzungen, Annahmen oder Extrapolation gewonnen werden. Die Bewertung von Modellen bezieht sich auf den aktuellen Stand der Modelle.

Alle Ausprägungen können Aussagen über eine Architektur, aber auch über mehrere Architekturen desselben Systems treffen. Die Anzahl der Varianten des Funktionsnetzes beispielsweise lässt sich aus dem Funktionsnetz allein ablesen. Die Ausführungszeit einer Funktion ist jedoch nur mithilfe des Funktionsnetzes, der Softwarearchitektur sowie der Hardwarearchitektur zu bemessen. Das Vorgehen der Bewertung ändert sich nicht signifikant, sofern Metriken zur Verfügung stehen, die mehrere Modelle berücksichtigen. Durch die Vielzahl der beteiligten Modelle ist es jedoch möglich, dass in sich konsistente Architekturen durch die Verknüpfung miteinander logisch inkonsistent werden. Diese Inkonsistenz kann durch entsprechende Bewertungsverfahren sowohl des Systems als auch der Modelle ermittelt und ausgegeben werden.

Zusätzlich existieren Abhängigkeiten zwischen den verschiedenen Bewertungen. So kann die Voraussetzung für eine Bewertung des Systems sein, dass die Modelle einen gewissen Reifegrad erlangt haben. Ein Beispiel wäre Aussagen über die Flashzeit, die anhand der Codegröße ermittelt wird. Die Metriken für die Ermittlung der Codegröße können erst eingesetzt werden, wenn die dafür nötigen Werte in das Modell eingetragen wurden, der Reifegrad des Modells also entsprechend ist.

3.2.3 Zeitpunkt der Bewertung

Ob ein- und dieselbe Architekturbewertung in frühen oder späten Phasen vorgenommen wird, hat erheblichen Einfluss auf das Ergebnis der Bewertung. Bereits in frühen Phasen des Entwicklungsprozesses ist es von Interesse, das zu entwickelnde System zu bewerten. Oft sollen Machbarkeit, Produzierbarkeit oder Kosten so früh wie möglich ermittelt werden. Dabei wird in Kauf genommen, dass die Ergebnisse noch unscharf und fehlerhaft sind, sofern sich der Fehler bestimmen lässt.

Das heißt, in frühen Phasen wird die Bewertung einer Architektur hauptsächlich auf Schätzungen beruhen. Die der Bewertung zugrundeliegenden Architekturen sind noch nicht ausgereift, noch nicht spezifiziert genug, um genaue Aussagen zu ermöglichen. Dafür stehen aus vorigen Entwicklungen genügend Informationen zur Verfügung, um aussagekräftige Werte aus Schätzungen oder Erfahrungswerten zu liefern. Je genauer geschätzt werden kann, desto besser wird die frühe Bewertung sein. Das heißt, ein Ziel der frühen Bewerter ist es, möglichst genaue Schätzmetriken zu benutzen, die auch mit wenigen und unvollständigen Informationen gute Werte liefern.

In späten Phasen einer Entwicklung werden die Informationen genauer und vielfältiger, die Modelle werden mit Informationen gefüllt und so genau wie möglich spezifiziert. Die Bewertungen, die jetzt durchgeführt werden, sind sehr genau, basieren auf vielen genauen Informationen und greifen nicht mehr auf Schätzungen zurück. Die Bewertungen greifen aber auf dieselben Modelle zurück, wie die Bewertungen der frühen Phasen.

Das bedeutet für die Architekturmodelle, dass sowohl die Informationen für die frühen als auch für die späten Bewertungen zur Verfügung gestellt werden müssen. Abstrakte Modelle mit wenigen, unterspezifizierten Informationen müssen durch Informationszuwachs in genaue Modelle überführt werden können, die als Grundlage z. B. der Codegenerierung (für Softwarearchitekturen) dienen können. Diese Anforderungen an Architekturmodelle müssen bei deren Design berücksichtigt werden. Die Architekturmodelle des VEIA-Referenzprozesses sind solche Modelle, die sowohl unterspezifiziert Aussagen ermöglichen, als auch detailliert beschrieben und weiterverwendet werden können. Das heißt wiederum, der Abstraktions- und Spezifikationsgrad ein- und desselben Architekturmodells variiert je nach Prozessphase der Entwicklung.

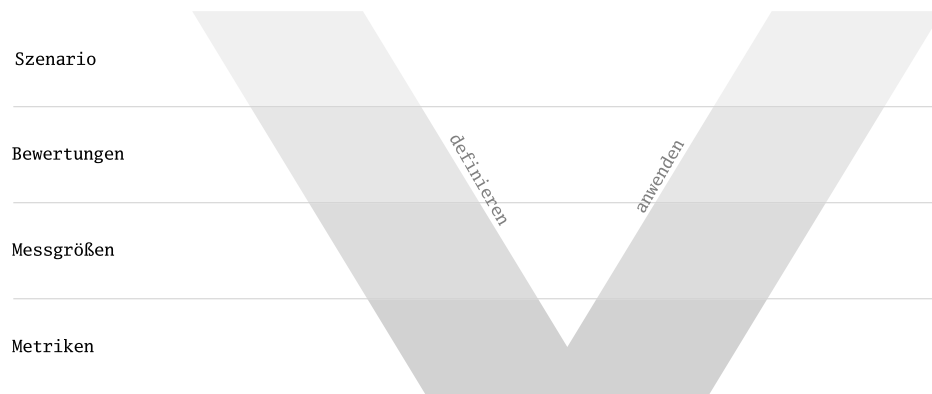
3.3 Informationsgehalt und Nutzen von Architekturmodellen

Eine zu beantwortende Frage ist, welche Informationen in Architekturmodellen erfasst und vorgehalten werden müssen, um Architekturbewertungen durchführen zu können. Das Ziel der Bewertung gibt an, welche Messgrößen für die Bewertung benötigt werden. Der Gegenstand der Bewertung gibt an, welche Architekturen in welchem Sinn bewertet werden müssen. Der Zeitpunkt der Bewertung gibt an, welche Art von Metriken für die Ermittlung der Messwerte eingesetzt werden sollen.

Aus diesen Anforderungen an die Bewertung kann für jeden einzelnen Punkt bestimmt werden, welche Informationen von den Architekturen zur Verfügung gestellt werden müssen, um die Metrik einzusetzen, die Messwerte zu bestimmen und damit die Bewertung durchzuführen. Das heißt wiederum, dass jede Architektur einen bestimmten Nutzen für die Architekturbewertung hat, je nach Information, die sie zur Bewertung beisteuert. Die Softwarearchitektur beispielsweise steuert zur Bewertung der Varianz des Funktionsnetzes nichts bei, während sie für die Entscheidung, ob 150 %-Komponenten gebaut werden, unerlässlich ist.

Der Nutzen einer Architektur für die Bewertung lässt sich also erst am konkreten Ziel der Bewertung bestimmen. Die Aufteilung einer Bewertung auf die zugrundeliegenden Messgrößen und Metriken kann die Nutzenabschätzung aber

Abbildung 3 Vorgehen bei der Architekturbewertung.



vereinfachen, indem ausgehend von den Metriken Klassen von Bewertungen gebildet werden, die bestimmte Informationen brauchen. Für diese Klassen von Bewertungen lassen sich dann Nutzendiskussionen einzelner Architekturen leichter führen. Die Klassifizierung von Bewertungen, Messgrößen und Metriken wird in den folgenden Phasen von VEIA vorgenommen.

3.4 Vorgehen

Eine Architekturbewertung ist ein Prozess, an dessen Ende die Bewertung einer Architektur steht. Dabei werden im Laufe der Bewertung verschiedene Fragen beantwortet, dafür Kriterien definiert, die über Metriken gemessen werden usw. Dieses Kapitel versucht, ein allgemeines Vorgehen für die Architekturbewertung zu finden, zum einen, um die verschiedenen Elemente der Architekturbewertung zu motivieren, zum anderen, um einen Leitfaden zu geben, wie bei einer Architekturbewertung vorgegangen werden kann. [Abbildung 3](#) zeigt eine schematische Darstellung des Vorgehens.

Die einzelnen Schritte werden im Folgenden kurz vorgestellt. Sie sind stark an die Elemente der Architekturbewertung gekoppelt, die in den folgenden Kapiteln detailliert vorgestellt werden.

Szenario aufstellen

Eine Architekturbewertung beginnt mit der Aufstellung des Bewertungsszenarios. Zuerst wird die Frage formuliert, die im Rahmen der Bewertung beantwortet werden soll, die Beantwortung der Frage ist Ziel des Szenarios.

Bewertungen bestimmen

Zur konkreten Frage werden Kriterien gesucht, die geeignet sind, eine Antwort zu liefern. Für die gefundenen Kriterien werden Bewertungen gesucht, das heißt, es wird festgelegt, wie die Kriterien zu bewerten sind, um eine Antwort auf die eingangs gestellte Frage zu finden. Es wird ebenfalls festgelegt, welche Architekturen Untersuchungsgegenstand werden sollen.

Messgrößen bestimmen

Zu den gefundenen Bewertungen werden jetzt die Messgrößen gesucht, deren Werte Grundlage für die Bewertungen sind. Für jede Bewertung können eine oder mehrere Messgrößen benötigt werden.

Metriken bestimmen

Aus den gesuchten Messgrößen und den zu verwendenden Architekturen können die Metriken ermittelt werden, die auf die Architekturen angewendet werden müssen. Auch hier können eine oder mehrere Metriken für eine Messgröße gefunden werden.

Rechter Ast von [Abbildung 3](#)

Alle Vorbereitungen sind abgeschlossen, jetzt müssen die Metriken auf die Architekturen angewendet werden. Die erhaltenen Messgrößen werden bewertet, die einzelnen Bewertungen werden in Zusammenhang gebracht, um die Eingangsfrage zu beantworten. Das Szenario ist komplett abgearbeitet.

4 Szenarios und Ziele

Welche Frage soll beantwortet werden?

Eine Architekturbewertung hat ein Ziel, das durch eine Frage, die Zielfrage, charakterisiert wird. Es wird die Qualität des modellierten Systems geprüft, es wird untersucht, ob das modellierte System machbar ist, es wird sichergestellt, dass das modellierte System echtzeitfähig ist usw.

Für die Qualitätsprüfung könnte die Frage »Ist das Produkt qualitativ hochwertig genug für unsere Produktpalette?« lauten. Eine andere Frage wäre: »Ist das modellierte System unter den monetären Randbedingungen machbar?«

Um die Frage zu analysieren, Kriterien für ihre Beantwortung anzugeben, Metriken herauszusuchen, kurz, um die Frage zu beantworten, erschaffen wir ein *Szenario* (angelehnt an [CMC05]). Das Szenario umfasst eine konkrete Architekturbewertung, die auf die Beantwortung einer bestimmten Frage abzielt. Das Szenario bündelt gewissermassen die Elemente der Architekturbewertung.

Ein Szenario ist als Hilfskonstrukt sinnvoll, da Bewertungen, Metriken, Messgrößen etc. zur Genüge existieren, aber nur in einem konkreten Zusammenhang sinnvoll eingesetzt werden können. Das Kriterium »Codegröße« ist z. B. bei dem Szenario »Echtzeitbewertung« anders zu wichten als beim Szenario »Machbarkeit«. Das gleiche Kriterium kann auch in einer frühen Bewertung andere Bedeutung besitzen als in einer späten Bewertung. Daher sind die drei Unterscheidungen der Architekturbewertung: Zweck, Gegenstand und Zeitpunkt der Bewertung, für jedes Szenario neu zu bestimmen und im Szenario festzuhalten.

Ein Szenario dient nicht nur dazu, die Elemente der Bewertung zu verwalten, sondern insbesondere dazu, aus der gegebenen Frage heraus die zur Beantwortung notwendigen Elemente herauszufinden und in einen Zusammenhang zu stellen. Dabei muss zunächst die Frage dahingehend analysiert werden, welche Kriterien zu ihrer Beantwortung nötig sind. Gleichzeitig muss bestimmt werden, welche Architekturen zur Beantwortung der Frage herangezogen werden. Aus diesen Informationen können die notwendigen Bewertungen gezogen werden, die die Kriterien repräsentieren.

4.1 Beispielfragen

In diesem Abschnitt werden Beispiele für Zielfragen vorgestellt, um einen Einblick in die Vielfalt der möglichen Szenarios zu geben.

Bewertung von Systemeigenschaften

- Ist das modellierte System machbar?
- Wie stabil ist der Hardwarearchitekturentwurf gegenüber Änderungen der Softwarearchitektur?
- Kann die zugeteilte Ressource die Anforderungen erfüllen?
- Entspricht mein Modell der Anforderung? (Verifikation)

Bewertung von Produktlinieneigenschaften

- Wieviele Gleichteile kann ich verbauen?
- Sind zwei Funktionen ähnlich genug, um als Varianzpunkt mit zwei Varianten modelliert zu werden?
- Soll eine Produktlinie mit speziellen Komponenten oder mit universellen Komponenten gebaut werden?
- Wie viele Abhängigkeiten zwischen Varianzpunkten gibt es?
- Gibt es Varianzpunkte nur auf einer Ebene oder auf mehreren?

Bewertung von Prozesseigenschaften

- Wieviel Aufwand wäre es, auf eine Produktlinie umzustellen vs. Beibehaltung der einzelnen Produkte?
- Wie groß ist der Testaufwand für die Softwarekomponenten der Produktlinie?
- Welchen Entwicklungsaufwand würde die Einführung einer neuen Funktionalität verursachen?
- Welche Kosten entstünden durch die Verwaltung und Verteilung einer neuen Softwarekomponente auf Werk und Werkstätten?

Bewertung von Modelleigenschaften

- Ist meine Modellierung mittlerweile zu komplex geworden?
- Sind meine Funktionen auf gleichem Niveau modelliert oder sind große Unterschiede zu erkennen?

- Können alle modellierten Varianten auch tatsächlich mindestens einmal ausgeleitet werden?

4.2 Beispiel: Produktlinie oder Produkt für CBS

Das konkrete Fallbeispiel in diesem und den folgenden Kapiteln bezieht sich auf die Funktion Condition Based Service (CBS). CBS wird hier nicht näher erläutert, es ist ausführlich in [GEKM07, S. 39 ff.] beschrieben. Das Fallbeispiel wird in diesem Dokument begonnen und dient dazu, die Architekturbewertungselemente zu erläutern. Es ist inhaltlich noch kein reales Beispiel, diese Arbeit wird in den folgenden Projektphasen durchgeführt, so dass von VEIA die unten aufgeworfene Frage für CBS beantwortet werden kann.

Die Zielfrage für das Fallbeispiel lautet:

Ist es günstiger, für CBS eine Software-Produktlinie mit für jedes Fahrzeug angepassten Softwarekomponenten zu bauen oder universelle Softwarekomponenten, die in jedem Fahrzeug parametrisiert werden müssen?

Der Gegenstand der Bewertung ist also eine Produktlinie, deren Lösungsalternativen miteinander verglichen werden sollen. Es sind zwei zu testende Lösungsalternativen angegeben: die 150 %-Lösung oder individuelle Softwarekomponenten für jedes Fahrzeug, die in eine einheitliche Softwarearchitektur eingebunden werden. Der Zeitpunkt der Bewertung ist in der Frage nicht angegeben, im Folgenden gehen wir davon aus, dass wir in einer mittleren Phase des Prozesses sind: Funktionsnetz, Software- und Hardwarearchitektur sind bereits definiert, die Entscheidung über die Softwarelösung soll gefällt werden.

Jetzt müssen die Kriterien und Architekturen bestimmt werden, die zur Beantwortung der Frage benötigt werden. Für den Hersteller ist wichtig, dass die Softwarekomponenten schnell flashbar sind. Der Verwaltungsaufwand soll minimal sein, die Kosten der Implementierung sollen den Nutzen bei der Flashbarkeit nicht übersteigen. Außerdem soll der Testaufwand minimiert werden.

Aus diesen Anforderungen lassen sich nun die Kriterien ableiten, die untersucht werden sollen:

- Flashzeit
- Verwaltungsaufwand

- Testaufwand
- Implementierungskosten

Die Architekturen, die zur Bewertung herangezogen werden, sind:

- Funktionsnetz,
- Softwarearchitektur,
- Hardwarearchitektur und
- Funktions- und Softwareverteilung.

5 Kriterien und Bewertungen

Was bedeuten die Messergebnisse?

Das Bewertungsszenario liefert die Kriterien, anhand deren die Zielfrage beantwortet werden soll. Die Werte für diese Kriterien müssen aus den Modellen gewonnen und in Beziehung zu anderen Werten gesetzt, bewertet werden. Für die Bewertung müssen mit Hilfe von *Metriken* die Werte der *Messgrößen* ermittelt werden, die im Rahmen der Bewertung interpretiert und beurteilt werden, um die Bedeutung des *Messwerts* in Relation zu Referenzwerten oder anderen Werten zu setzen. Vereinfacht gesagt: eine Bewertung nutzt *Metriken*, um die Werte einer Messgröße zu messen und zu vergleichen.

Für die Bewertung eines Kriteriums müssen die relevanten Messgrößen und die Vergleichsmaßstäbe ermittelt oder festgelegt werden. So kann z. B. das Kriterium »Kosten« aus den Messgrößen »Entwicklungskosten« für Softwarearchitekturen und »Bauteilkosten« für Hardwarearchitekturen zusammengesetzt werden. Sind die Messgrößen bestimmt, muss ein Maßstab festgelegt werden, anhand dessen die Messgrößen eine Bedeutung bekommen. Dieser Maßstab kann relativ zu anderen Werten sein oder relativ zu einem Referenzwert. Die so gewonnene Einschätzung der Messwerte ist die Bewertung des Kriteriums.

Je nach Zweck, Gegenstand und Zeitpunkt der Bewertung sowie nach betrachteter Architektur können unterschiedliche Messgrößen für die Bewertung eines Kriteriums herangezogen werden. So kann für das Kriterium »Komplexität« in frühen Phasen die Messgröße »Funktionskomplexität« gemessen werden, in späten Phasen die »Codekomplexität«. Wenn aber die Messgrößen gleich bleiben sollen, also die »Komplexität« mit Hilfe der Messgröße »Codekomplexität« bestimmt werden soll, so können unterschiedliche Metriken eingesetzt werden. Im Beispiel könnte in frühen Phasen eine Schätzmetrik aufgrund von Erfahrungswerten benutzt werden und in späten Phasen eine Metrik, die über den Quellcode definiert ist. Je nach gewählter Variante sind Teile des Szenarios unabhängig von unterschiedlichen Bewertungsarten und können wiederverwendet werden.

Die Bewertung kann nur dann korrekt durchgeführt werden, wenn die Güte der Messwerte berücksichtigt wird. Die Güte der Messwerte hängt von den verwendeten Metriken ab. Wird eine Schätzmetrik benutzt, sind die Ergebnisse anders zu bewerten als mit Messmetriken. Auch die Genauigkeit der Messmetriken kann unterschiedlich sein, je nach Rechenaufwand, einbezogenen Attributen etc.

5.1 Beispielkriterien und deren Bewertungen

- Speicherbedarf: die Architektur mit dem größeren Speicherbedarf ist die schlechtere
- Flashzeit: die Architektur mit der kleineren maximalen Flashzeit ist die bessere
- Komplexität: zwei Architekturen haben ähnliche Komplexität, wenn ihre jeweilige Funktionskomplexität nicht mehr als 10 % voneinander abweicht

5.2 Beispiel: Produktlinie oder Produkt für CBS

Die Kriterien, die bestimmt wurden, sind:

- Flashzeit
- Verwaltungsaufwand
- Testaufwand
- Implementierungskosten

Die Kriterien werden gewichtet und in Relation gebracht, um aus den unterschiedlichen Ergebnissen eine Bewertung zu bilden und damit die Frage beantworten zu können:

- 1 Flashkosten einer Lösung = Flashzeit * Minutensatz
- 2 Gesamtaufwand = Wichtung 1 * Verwaltungsaufwand + Wichtung 2 * Testaufwand
- 3 Gesamtkosten = Wichtung 3 * Flashkosten + Wichtung 4 * Implementierungskosten

Die Flashkosten sind das Produkt von Flashzeit und Minutensatz. Die Flashkosten selbst sind noch keine Bewertung, sondern ein Zwischenwert, der für die Bewertung benötigt wird. Da die Flashkosten aber eine zusammengesetzte oder abgeleitete Messgröße sind, deren Wert sich durch Kombination von den Werten elementarer Messgrößen ergibt, zählt die Berechnung der Kosten zu den Aufgaben der Bewertung. Die Ermittlung elementarer Messgrößen wie Flashzeit ist Aufgabe von Metriken. Für Gesamtaufwand und Gesamtkosten ergeben sich analoge Betrachtungen.

Nach der Berechnung der Zwischenwerte werden zwei Entscheidungen getroffen: Eine Architektur wird beim Aufwand bevorzugt, wenn ihr Gesamtaufwand geringer ist. Analog wird eine Architektur bei den Kosten bevorzugt, wenn ihre

Gesamtkosten geringer sind. Aus diesen beiden Einzelentscheidungen wird die Antwort auf die Zielfrage gebildet: Eine Architektur gilt als günstiger, wenn Aufwand und Kosten zu ihren Gunsten entschieden wurden. Trifft das auf keine der Architekturen zu, sind Gesamtaufwand und Gesamtkosten im Verhältnis 2:3 zu einem Wert zu addieren und zu vergleichen. Die Architektur mit dem kleineren Gesamtwert ist günstiger.

Es ist also klar definiert, welche Bewertungen auf Basis welcher Architekturen durchzuführen sind und wie die Ergebnisse der Bewertungen zu wichten und zu verbinden sind, um zu einer Gesamtentscheidung zu gelangen.

6 Messgrößen und Messwerte

Was wird gemessen?

Eine Messgröße ist eine Eigenschaft einer Architektur, die über einen Wert, den Messwert, gemessen werden kann. Eine Messgröße kann durchaus für mehrere Architekturen definiert sein, z. B. kann die »Komplexität« für Funktionsnetze, für Software- oder für Hardwarearchitekturen bestimmt werden. Der Wert einer Messgröße wird durch eine Metrik bestimmt.

Um die Vergleichbarkeit von Messwerten zu erreichen, müssen die Messwerte normiert werden, das heißt, vergleichbar gemacht werden. Diese Normierung kann entweder im Rahmen der Bewertung stattfinden oder bereits durch eine Metrik vorgenommen werden.

6.1 Beispielmessgrößen

- Softwarekomponenten-Größe
- Implementierungsaufwand jeder Softwarekomponente
- Ausführungszeit einer Softwarekomponente
- maximale Buslast
- Ähnlichkeit zweier Funktionen
- Kosten

6.2 Beispiel: Produktlinie oder Produkt für CBS

Die Messgrößen für die definierten Kriterien müssen festgelegt werden:

- Flashzeit: Softwarekomponenten-Größe
- Verwaltungsaufwand: Anzahl der Softwarekomponenten sowie Verwaltungsaufwand für eine Softwarekomponente
- Testaufwand: Anzahl der Testfälle sowie Aufwand pro Testfall
- Implementierungskosten: Implementierungsaufwand jeder Softwarekomponente

7 Metriken

Wie wird gemessen?

Metriken sind der letzte betrachtete Baustein im Gefüge der Architekturbewertung. Eine Metrik gibt an, wie der Wert einer Messgröße bestimmt wird, das heißt, eine Metrik ist eine Funktion, die eine Eigenschaft einer Architektur in einen Wert abbildet. Wie bereits im Kapitel über Messgrößen dargelegt, sind Metriken architekturabhängig. Das heißt, eine Metrik gilt nur für eine bestimmte Architektur oder Kombination von Architekturen.

Je nach Architektur, Zeitpunkt, Gegenstand und Zweck werden unterschiedliche Metriken benutzt, um den Wert einer Messgröße zu bestimmen. Die »Komplexität« kann z. B. in Funktionsnetzen durch Zählung der Ein- und Ausgänge ermittelt werden. Für Softwarearchitekturen stehen Codemetriken bereit, die die Anzahl der Verzweigungen oder Schleifen ermitteln.

Zusätzlich zu den genannten Unterscheidungsmerkmalen – Architektur, Zeitpunkt, Gegenstand und Zweck – können Metriken noch anhand der Rechenmethode – Schätzen oder Berechnen – und anhand der Messqualität unterschieden werden. Rechenmetriken können auf Architekturen angewendet werden, bei denen genaue Informationen vorliegen, Informationen, die mathematisch durch die Metriken erfasst werden können. Dies ist meist in späten Entwicklungsphasen der Fall, so dass Rechenmetriken oft in späten Entwicklungsphasen vorkommen. Schätzmetriken ermitteln ihren Wert aufgrund von Schätzungen, die auf Erfahrungswerten oder angenommenen Referenzwerten basieren. Sie werden dann bevorzugt eingesetzt, wenn die zur Verfügung stehenden Informationen noch zu ungenau für berechnende Metriken sind.

Die Qualität oder Güte einer Metrik hängt neben der Unterscheidung zwischen Schätzen und Messen von den verwendeten Informationen ab, die in die Metrik einfließen. Bei vielen Metriken müssen Aufwand und Genauigkeit gegeneinander abgewogen werden. So können Echtzeitmetriken sehr genau, aber sehr langsam implementiert werden, oder ungenauer, dafür schneller. Die Qualität einer Metrik ist für die Bewertung von entscheidender Bedeutung, da die Messergebnisse nur unter Berücksichtigung ihrer Entstehung korrekt interpretiert werden können.

Metriken können über einzelne Architekturen, aber auch über Kombinationen von Architekturen definiert sein. So ist die Komplexität des Funktionsnetzwerks eine

Eigenschaft nur dieser Architektur. Der Ressourcenverbrauch eines Funktionsnetzwerks ist aber nur im Zusammenhang mit der Softwarearchitektur bestimmbar, da erst die Softwarekomponenten Informationen über die benötigten Ressourcen enthalten. In frühen Entwicklungsphasen kann der Ressourcenverbrauch anhand historischer Werte vorangegangener Softwarearchitekturen geschätzt werden, in späten Entwicklungsphasen kann er genau berechnet werden.

Das heißt, die Messgröße »Ressourcenverbrauch« kann durch eine Schätzmeterik, aber auch durch eine Berechnungsmeterik gemessen werden. Bei guten Schätzmeteriken liegen beide Werte einigermaßen nah beieinander.

Metriken, die mehrere Architekturen benötigen, können auch auf Informationen aller beteiligten Architekturen zurückgreifen. Eine Wort-Case-Ausführungszeit-Berechnung benötigt Informationen der Softwarearchitektur, der Hardwarearchitektur sowie des Verteilungsmodells zwischen Software- und Hardwarearchitektur.

Metriken über Produktlinien bilden einen weiteren Sonderfall. Im Gegensatz zu Metriken, die den Wert einer Messgröße aus einer Architektur bestimmen, müssen Metriken über Produktlinien den Wert der Messgröße aus mehreren Lösungen der Produktlinie bestimmen. Je nach Lösungsraum müssen die Werte der einzelnen Lösungen bestimmt und miteinander in Relation gesetzt werden. Je nach Metrik kann das einfach sein oder komplexe Annahmen erfordern. Eine einfache Metrik über Produktlinien ist der Speicherbedarf: hier wird der Speicherbedarf für jede Lösung bestimmt und das Maximum als Speicherbedarf der Produktlinienarchitektur ausgegeben.

Gütekriterien können u. a. sein:

- Objektivität
- Normierung
- Vergleichbarkeit

7.1 Beispielmetriken

- Zählung der Ein- und Ausgangsports einer Funktion
- Messung der Codegröße
- Function-Point-Verfahren
- Halstead-Metrik
- Minimum des Bandbreite aller modellierten Busse

7.2 Beispiel: Produktlinie oder Produkt für CBS

Die Metriken für die Messgrößen des Szenarios sind:

- SWC-Größe: Messung der maximalen SWC-Größe
- Anzahl der SWC: Zählung der Softwarekomponenten
- Verwaltungsaufwand einer SWC: Anzahl der SWC * Verwaltungskosten je SWC
- Anzahl der Testfälle: Zählung der Testfälle
- Aufwand pro Testfall: Maximum des Aufwands der Testfälle
- Implementierungsaufwand jeder SWC: Schätzung aufgrund voriger Projekte + 10 %

8 Einbettung in den Entwicklungsprozess

Die Architekturbewertung ist ein fester Bestandteil jedes Entwicklungsprozesses. Fragen über das modellierte System oder die Güte der Modellierung sind notwendig, um den Entwicklungsprozess zielgerichtet durchführen zu können.

In diesem Dokument wurden hauptsächlich Fragen über das modellierte System oder über die Modelle behandelt. Darüber hinaus gehören zu einer Architekturbewertung auch Fragen über begleitende oder unterstützende Prozesse. So können beispielsweise Werkzeuge, Kompetenzen oder Investitionen für die Architekturbewertung herangezogen werden. Eine typische Frage ist: »Inwieweit ändert sich die Zertifizierung der Werkzeuge, wenn eine Produktlinie mit individuellen Softwarekomponenten verwendet wird?« Eine weitere Frage ist: »Ist die Entscheidung für oder gegen eine Produktlinie mit der Lieferantenschnittstelle kompatibel?« Solche Fragen wurden in dieser Projektphase noch nicht untersucht.

Die weiteren Arbeiten sehen vor, das Fallbeispiel *Condition Based Service* (CBS) weiter auszuformulieren. Es soll eine umfassende Architekturbewertung vorgenommen werden. Dazu werden Szenarios für die Modellbewertung definiert, aber auch Szenarios betrachtet, die die eben erwähnten begleitenden Prozesse betreffen.

Literatur

- [CMC05] *Clements, Paul C. ; McGregor, John D. ; Cohen, Sholom G.:* The Structured Intuitive Model for Product Line Economics (SIMPLE) / CMU-SEI. – Technical Report CMU/SEI-2005-TR-003. <http://www.sei.cmu.edu/publications/documents/05.reports/05tr003.html>
- [GEKM07] *Große-Rhode, Martin ; Euringer, Simon ; Kleinod, Ekkart ; Mann, Stefan:* Grobentwurf des VEIA-Referenzprozesses / Fraunhofer-Institut für Software- und Systemtechnik, Abt. Verlässliche Technische Systeme. Mollstraße 1, 10178 Berlin, Germany, Januar 2007. – ISST-Bericht 80/07. Projekt VEIA. <http://veia.isst.fraunhofer.de/>