

Qualität von Unternehmensarchitekturen

Seminararbeit

im Modul Strategisches Informationsmanagement
Masterstudiengang Information Systems and Services
Hochschule der Medien Stuttgart

Onur Birsöz, B.Sc.
Hannes Meyer, B.Sc.

Betreut durch Prof. Dr. B. Schwarzer

Stuttgart, Januar 2009

Inhaltsverzeichnis

| | |
|---|-----------|
| Inhaltsverzeichnis | 2 |
| 1 Einleitung..... | 3 |
| 2 Begriffsdefinitionen und Abgrenzung..... | 4 |
| 2.1 Qualität | 4 |
| 2.2 Unternehmensarchitektur..... | 4 |
| 3 Instrumente zur Messung der Qualität von IT-Systemen | 5 |
| 3.1 Benchmarking..... | 5 |
| 3.2 Prozesskostenrechnung..... | 6 |
| 3.3 Target Costing | 6 |
| 3.4 Balanced Scorecard..... | 6 |
| 4 Qualitätsmaßstäbe einer Unternehmensarchitektur..... | 7 |
| 4.1 Übergreifende Ansprüche | 7 |
| 4.1.1 Agilität..... | 7 |
| 4.1.2 Integration..... | 8 |
| 4.2 Qualitätsansprüche aus der Softwareentwicklung..... | 9 |
| 4.2.1 Produktbezogene Aspekte | 10 |
| 4.2.2 Nutzerbezogene Aspekte | 11 |
| 4.2.3 Prozessbezogene Aspekte..... | 12 |
| 5 Metriken für Unternehmensarchitekturen..... | 14 |
| 5.1 Einleitung..... | 14 |
| 5.2 Definition einer Metrik | 14 |
| 5.3 Graphen..... | 15 |
| 5.3.1 Senke..... | 15 |
| 5.3.2 Quelle | 15 |
| 5.3.3 Gewicht..... | 15 |
| 5.4 Arbeitslast..... | 16 |
| 5.5 Zeiten und Latenz | 17 |
| 5.6 Entropie | 19 |
| 5.7 Skalierbarkeit..... | 20 |
| 5.8 Zusammenfassung | 22 |
| 6 Zusammenfassung und Ausblick..... | 23 |
| Literaturverzeichnis | 24 |

1 Einleitung

Die Betrachtung der Qualität eines Produkts, einer Dienstleistung, Systems oder von Prozessen ist von großer Bedeutung. Im Produkt- und Dienstleistungsbereich bestehen daher zahlreiche Ansätze zu Messung und Steuerung der Qualität. Die Messung ist dabei eine Grundvoraussetzung zur Verbesserung der Qualität. Im IT-Bereich existieren bereits Kennzahlen-basierte Ansätze zur Messung von Qualität, auch im Hinblick auf größere Software- und Hardwaresysteme (vgl. z.B. Malich 2008). Im Bereich der Unternehmensarchitekturen besteht hingegen noch deutlicher Forschungsbedarf. Zwar wurden bereits monetäre Bewertungsmodelle erarbeitet (vgl. vom Brocke et al. 2008), zur ganzheitliche Qualitätsbetrachtung mit Hilfe von Metriken existiert jedoch noch kein umfassender Standard. Dabei ist die Unternehmensarchitektur für den Betrieb eines Unternehmens von großer Bedeutung, da sie Einfluss auf die Effizienz und Effektivität des gesamten Business haben kann.

In dieser Arbeit werden daher unterschiedliche Qualitätsmaßgrößen untersucht, die im Ganzen einen Einfluss auf die Qualität einer Architektur haben können. Es wird versucht, diese so umfassend wie möglich zu beschreiben und dabei alle wichtigen Aspekte zu berücksichtigen. Darüber hinaus werden Metriken untersucht, mit denen auf mathematische Weise eine Qualitätsbewertung ermöglicht werden kann. Aufgrund der spärlichen Behandlung des Themas in der Literatur werden dabei auch Aspekte aus der Softwareentwicklung übernommen und diese auf das Thema der Unternehmensarchitekturen übertragen.

Diese Arbeit gliedert sich in die Teilkapitel Begriffsdefinitionen, in dem die Begriffe Qualität und Unternehmensarchitektur knapp beschrieben werden, einer Sammlung der Qualitätsmaßstäbe einer Unternehmensarchitektur und einer Beschreibung der auf mathematischen Methoden basierenden Metriken zur Qualitätsbewertung.

2 Begriffsdefinitionen und Abgrenzung

2.1 Qualität

Qualität ist nach der Definition der Norm ISO 9000:2005 der „Grad, in dem ein Satz inhärenter Merkmale Anforderungen erfüllt“. Diese Definition macht deutlich dass es sich bei Qualität um eine Zusammensetzung aus mehreren Einzelmerkmalen handelt (ein Satz von Merkmalen) und dass diese Merkmale dem Objekt innewohnen (inhärent sind), ihm also nicht von außen zugeordnet werden. Erfolgt die Messung der Qualität ausschließlich anhand dieser inhärenten Merkmale, so werden subjektive Aspekte wie Kundenzufriedenheit oder Auswirkungen nicht berücksichtigt. Da die Messung subjektiver Ansprüche nur sehr schwierig möglich ist, werden diese Ansprüche häufig in inhärente Merkmale transferiert und somit als Teilmerkmal eines Produktes betrachtet, wodurch eine Messung wiederum ermöglicht wird.

2.2 Unternehmensarchitektur

Für den Begriff Unternehmensarchitektur (Enterprise Architektur) existiert derzeit noch keine allgemeingültige und international anerkannte Definition. Nach der Definition von Aier/Dogan 2005 besteht die Unternehmensarchitektur aus der Organisationsarchitektur (bestehend aus Organisationsstruktur und Geschäftsprozessen) und der IT-Architektur (zu der unter anderem die Informationssystemarchitektur zählt). Demnach werden nicht-technische (Organisation) und technische Bestandteile (IT) in einer Architektur vereint. Architektur wird hierbei sowohl als der Plan zum Aufbau eines Systems als auch dessen spätere konkrete Ausprägung verstanden. Aier/Dogan stellen fest, dass in der Literatur keine scharfe Grenze zwischen IT und Organisation gezogen wird und manche Autoren bei der IT-Architektur auch organisatorische Aspekte betrachten und vice versa.

Zusammenfassend lässt sich festhalten, dass bei der Betrachtung von Unternehmensarchitekturen sowohl organisatorische als auch technische Aspekte zu berücksichtigen sind und diese weitestgehend untrennbar miteinander verwoben sind. Dies ermöglicht eine ganzheitliche Betrachtung der Problematiken und durch Abstimmung der Interessen der Teilbereiche den Aufbau einer an den Geschäftszielen ausgerichteten Informationssystemarchitektur.

3 Instrumente zur Messung der Qualität von IT-Systemen

In diesem Kapitel werden einige aktuelle Instrumente zur Unterstützung der Führungsprozesse im IT-Bereich dargestellt. Die Eignung zur Qualitätsmessung wird dabei kurz betrachtet, da es sich bei den Systemen um relativ allgemein gehaltene Konzepte handelt. Dieses Kapitel richtet sich stark nach Hagen/Schwinn 2006, da in diesem Beitrag die Führungsinstrumente knapp und themengerecht dargestellt wurden. Allerdings werden in diesem Beitrag die Instrumente nach der Eignung zum Einsatz bei der Integration bewertet, welche lediglich einen Teilaspekt von Unternehmensarchitekturen darstellt (vgl. Kapitel 4.1.2).

Zur Einführung werden die Begriffe Kennzahl und Kennzahlensystem definiert, da diese Konzepte bei der Messung im IT-Bereich eine bedeutende Rolle spielen.

Kennzahlen dienen der Zusammenfassung messbarer Größen, mit deren Hilfe Aussagen über den Zustand eines Systems ermöglicht werden können. Eine Kennzahl muss aus einem quantifizierbaren Sachverhalt bestehen, also auf einer Skala messbar sein. Weitere Ansprüche an eine Kennzahl sind wie folgt.

- Der Inhalt der Kennzahl muss dem Informationszweck entsprechen, um die Anforderung erfüllen zu können.
- Die Kennzahl muss so genau wie nötig sein, um Anforderungen an Reliabilität und Validität zu erfüllen.
- Der Zeitraum zwischen Erhebung und Auswertung einer Kennzahl muss möglichst gering sein, um eine größtmögliche Aktualität zu garantieren.
- Der Wert der Erkenntnisse, zu denen eine Kennzahl führt, sollte die Kosten zur Erhebung übersteigen (Kosten-Nutzen-Relation).
- Die Kennzahl muss vom späteren Verwender interpretiert werden können und eine Zurückverfolgung erlauben.

Eine gewisse Menge von Kennzahlen wird in einem *Kennzahlensystem* zusammengefasst. In diesem System befinden sich Kennzahlen, die sich gegenseitig beeinflussen. Die Betrachtung des gesamten Kennzahlensystems führt zu neuen Erkenntnissen über die gemessenen Sachverhalte.

3.1 Benchmarking

Beim Benchmarking wird die Leistung des Unternehmens in Teilbereichen mit anderen Unternehmen verglichen. Dabei wird zwischen quantitativem und qualitativem Benchmarking unterschieden. Beim quantitativen Ansatz soll die eigene Position im Wettbewerbsumfeld festgestellt werden um daraus strategische Ziele abzuleiten. Beim qualitativen Ansatz hingegen wird das Ziel verfolgt, von „Best Practices“ zu lernen und diese im eigenen Unternehmen einzusetzen. Vergleichsobjekte können dabei Strategien, Methoden, Prozesse oder Produkte darstellen. Der Vergleichspartner kann sowohl ein di-

rekter Konkurrent aus der gleichen Branche oder ein branchenfremdes Unternehmen, als auch eine interne Abteilung oder Geschäftsbereich sein.

Im Architekturbereich bestehen derzeit noch keine Benchmarks (vgl. Hagen/Schwinn 2006), sodass dieses Instrument in der Praxis noch nicht zur Qualitätsbeurteilung eingesetzt werden kann. Aufgrund der expliziten Möglichkeit, qualitative Aspekte zu vergleichen, ist es jedoch prinzipiell geeignet. Voraussetzung ist jedoch in jedem Fall die Definition aussagekräftiger Kennzahlen.

3.2 Prozesskostenrechnung

Die Prozesskostenrechnung dient der verursachungsgerechten Umlegung von Gemeinkosten anhand der Geschäftsprozesse. Dadurch entsteht eine höhere Kostentransparenz, das Kostenbewusstsein wird gesteigert und es werden Potenziale für Prozessoptimierungen offengelegt. Letzteres Argument lässt sich unter Umständen zur Bewertung einer Unternehmensarchitektur einsetzen, da hierdurch die Qualität der Geschäftsprozesse gemessen werden kann. Da dies jedoch nur einen Teilaspekt darstellt und die Einführung einer Prozesskostenrechnung mit enormem Aufwand verbunden ist, wird die Prozesskostenrechnung nicht als Instrument zur Qualitätsbewertung empfohlen.

3.3 Target Costing

Target Costing, deutsch Zielkostenrechnung, ist ein Instrument zur Kostenplanung eines Produktes oder einer Dienstleistung. Hierbei richten sich die möglichen Kosten des Produktlebenszyklus nach den für das Produkt am Markt erzielbaren Preisen. Da es sich um ein Marketinginstrument handelt und der Einfluss einer Architektur auf den Preis eines Produktes nicht messbar ist, kann es zur Qualitätsbeurteilung nicht eingesetzt werden.

3.4 Balanced Scorecard

Die Balanced Scorecard (BSC) ist ein weit verbreitetes Führungsinstrument, welches klassisch die Strategiebestimmung und –Kontrolle aus verschiedenen Perspektiven ermöglicht. Als Kennzahlen werden hierbei im Gegensatz zu anderen Modellen sowohl finanzielle als auch nicht-monetäre Größen definiert. Besonders die verschiedenen Perspektiven lassen die Entwicklung verschiedener Kennzahlen aus unterschiedlichen Bereichen zu. Dabei kann beispielsweise die Auswirkung von Prozessen auf die IT und umgekehrt gemessen werden. Zur ganzheitlichen Betrachtung der Unternehmensarchitektur ist sie durch ihre Flexibilität besonders geeignet. Je nach Einsatzgebiet lassen sich die Perspektiven und Kennzahlen anpassen. So kann die BSC dazu operationalisiert werden, die Qualität eines Systems aus unterschiedlichen Blickwinkeln zu beurteilen und zu messen.

4 Qualitätsmaßstäbe einer Unternehmensarchitektur

In diesem Kapitel werden die für die Qualität relevanten Aspekte einer Unternehmensarchitektur dargestellt.

4.1 Übergreifende Ansprüche

Die folgenden Ansprüche zielen auf die Unternehmensarchitektur im Allgemeinen ab. Um das Business optimal unterstützen zu können, sind diese Aspekte von herausragender Bedeutung.

4.1.1 Agilität

Die Fähigkeit zur Anpassung an neue Gegebenheiten ist in einer sich schnell wandelnden Welt zu einem treibenden Faktor geworden. Nach Dove et al. 1996 bezeichnet Agilität die Fähigkeit eines Unternehmens, sich erfolgreich an ein sich ständig wechselndes und unvorhersehbares Unternehmensumfeld anpassen zu können. Da der Wandel und die dadurch nötige Anpassung nur schwer beeinflussbar sind, ist es das Ziel, die Phasen der Anpassung möglichst kurz zu halten, um dadurch Wettbewerbsvorteile erzielen zu können. Die Unternehmensarchitektur muss daher ausreichend flexibel gestaltet sein, um den geplanten Wandel mitschreiten zu können. Auch die IT muss im Sinne des IT-Business-Alignments im Stande sein, den geänderten Anforderungen folgen zu können und die Geschäftstätigkeit weiterhin unterstützen zu können. Für die Bestimmung der Qualität einer Unternehmensarchitektur ist als die Anpassungsfähigkeit ein wichtiger Faktor.

Nach Schwinn/Winter 2005 ergibt sich die Agilität einer Architektur aus fünf strategischen Teilzielen. Im Folgenden werden diese Ziele erläutert.

Komplexitätsreduktion durch Desintegration

Um gewachsene Applikationslandschaften mit mehreren hundert Anwendungen steuern zu können, werden Domänen aus Applikationen und Informationsobjekten gebildet. Innerhalb dieser Domänen besteht weiterhin eine enge Kopplung zwischen Anwendungen, Domänenübergreifend wird eine lose Kopplung angestrebt. Ziel ist sowohl eine bessere Übersichtlichkeit, als auch die Möglichkeit, einzelne Domänen unabhängig weiter entwickeln zu können und somit auf neue Anforderungen anpassen zu können.

Optimale Kopplungsgrade

Der optimale Kopplungsgrad beschreibt das Maß der Integration zwischen Systemen. Bei einer engen Kopplung bestehen viele Schnittstellen, die bei Änderungen gepflegt werden müssen. Bei einer losen Kopplung müssen hingegen zur Laufzeit Verbindungen, eventuell über eine Middleware, hergestellt werden. Die Abwägung zwischen loser

und enger Kopplung ist schwierig, zumal die Messung der Vorteile, Nachteile und Auswirkungen zur Zeit noch nicht formal durchgeführt werden kann.

Redundanzvermeidung durch Wiederverwendung

Zur Vermeidung von Redundanzen wird angestrebt, dass bestimmte Funktionen nur einmal implementiert werden müssen und danach mehrmals von unterschiedlichen Systemen verwendet werden können. Dadurch lassen sich sowohl Entwicklungs- als auch Pflegekosten minimieren. Bei der Entwicklung der Services stellt sich allerdings die Frage der optimalen Granularität. Werden Services sehr allgemein gehalten, können sie häufiger wiederverwendet werden, der Aufwand zur Entwicklung und Pflege steigt jedoch. Spezialisierte Services lassen sich hingegen weniger häufig wiederverwenden. Vorteilhaft ist bei dieser Thematik, dass sich die Wiederverwendung einfach messen lässt, dadurch können Kennzahlen definiert werden, die zu einem gewissen Teil eine Aussage über die Qualität des Gesamtsystems zulassen.

Minimale Aufwände für Integrationsprojekte

Die Integration neuer Komponenten in ein bestehendes System verursacht unter Umständen einen hohen Aufwand. Für die Agilität der Gesamtplattform ist es allerdings notwendig, solche Projekte durchzuführen. Um die Kosten so niedrig wie möglich zu halten, muss die Integration möglichst einfach durchzuführen sein. Die Kosten dürfen dabei auch nicht unabhängig von der Komplexität des Integrationsprojektes betrachtet werden, um die Qualität der Integrationsfähigkeit messen zu können.

Minimierung der Infrastrukturkomplexität und –kosten

Um Integrationsaufwände gering zu halten bedarf es einer möglichst geringen Komplexität der Infrastruktur. Mit steigender Zahl unterschiedlicher Systeme und Tools steigt auch der personelle Aufwand der Integration. Auf der anderen Seite bedarf es, um neue Technologien implementieren zu können, einer gewissen Grundmenge bestehender Technologien. So lassen sich Workarounds vermeiden, die auftreten können, wenn neue Technologien von neuen Systemen vorausgesetzt werden. Im Hinblick auf die zukünftige Ausbaufähigkeit sind derartige Workarounds zu vermeiden. Die Komplexität der Infrastruktur kann also als Qualitätsmerkmal herangezogen werden, jedoch nicht ohne den Blick zusätzlich auf die zukünftige Variabilität zu werfen.

4.1.2 Integration

Der Grad der Integration zwischen im Unternehmen verwendeten Informations- und Anwendungssystemen, Prozessen und Daten kann als Maßgröße zur Bestimmung der Qualität herangezogen werden. Der optimale Integrationsgrad wird aus monetärer Sicht hierbei durch die Kosten für Anwendungssysteme und Schnittstellen bestimmt. Folgende Abbildung verdeutlicht diese Zusammenhänge (nach Winter 2006).

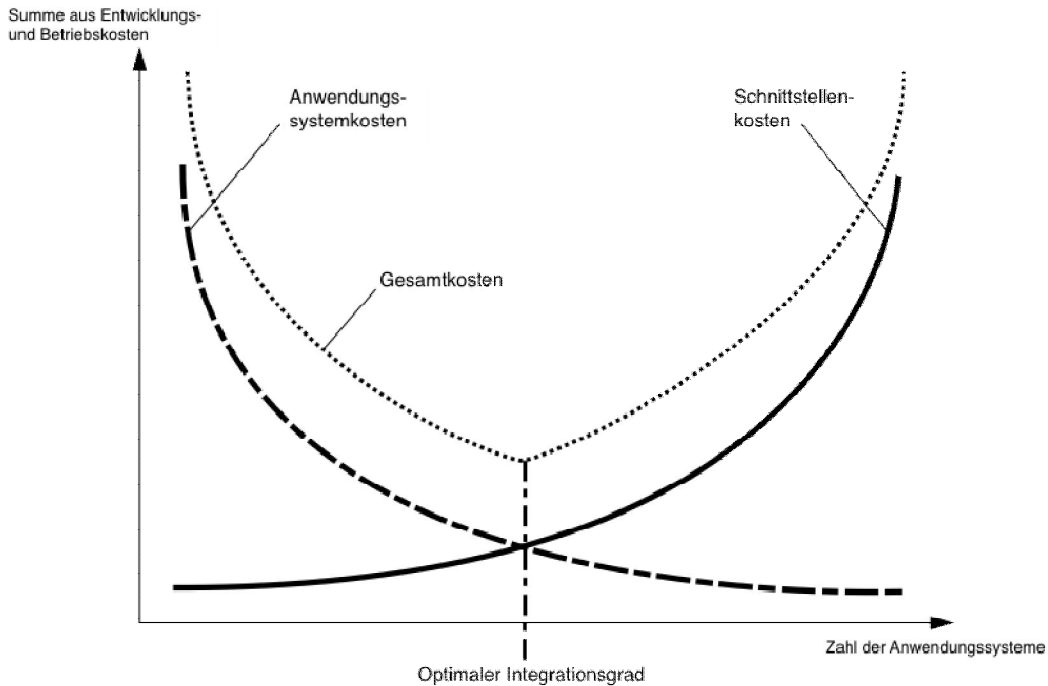


Abbildung 1: Integrationsgrad (nach Winter 2006)

Der hierbei angedeutete optimale Integrationsgrad ist allerdings kritisch zu betrachten, da er nur rein monetäre Ansprüche berücksichtigt, nicht jedoch die Auswirkungen auf die Agilität des Gesamtunternehmens. Eine größere Anzahl unterschiedlicher Systeme kann die Agilität des Gesamtunternehmens begünstigen, da bei Änderungen nur Teilsysteme betroffen sind und nicht große Systeme verändert werden müssen. Eine große Verflechtung zwischen unterschiedlichen Systemen führt dabei jedoch zu einer höheren Anzahl an Schnittstellen, die bei Änderungen unter Umständen ebenfalls gepflegt werden müssen. Ein für ein Unternehmen optimaler Integrationsgrad wird also durch mehr als nur Kosten beeinflusst, so werden hierbei beispielsweise die Befriedigung der Bedürfnisse der Stakeholder (vgl. nachfolgende Kapitel) nicht als Maßgröße berücksichtigt. Als Maßstab der Qualität einer Unternehmensarchitektur ist der Integrationsgrad alleine folglich nicht ausreichend, daher werden nachfolgend weitere Aspekte untersucht.

4.2 Qualitätsansprüche aus der Softwareentwicklung

Zur Qualität von Softwaresystemen und –architekturen bestehen bereits zahlreiche Untersuchungen. Diese unterscheiden im Wesentlichen produkt-, nutzer- und prozessbezogene Aspekte, welche in den folgenden Kapiteln näher beschrieben werden. (vgl. Malich 2008, Achtert 2005, sämtlich zitiert nach Garvin 1984). Malich referenziert besonders zwei Standards, in denen Qualitätsmerkmale von Softwarearchitekturen definiert sind: den ISO-Standard ISO/IEC 9126 (vgl. ISO 2001) und das Qualitätsmodell des Software Engineering Institute der Carnegie Mellon Universität Pittsburgh (vgl. Bass/Clements/Kazmann 2003). Im Folgenden werden die für Unternehmensarchitektu-

ren wichtigsten Merkmale dieser Modelle unter den drei oben genannten Aspekten zusammengefasst.

4.2.1 Produktbezogene Aspekte

Funktionalität

Hierunter fallen Kriterien wie die Fähigkeit, Aufgaben und Probleme angemessen behandeln zu können und dies auf eine präzise Art und Weise zu tun. Die Interoperabilität zwischen Systemen muss wenn nötig unterstützt werden, um einen an den Prozessen ausgerichteten Datenfluss zu ermöglichen und Medienbrüche zu vermeiden. Dabei muss das Gesamtsystem sicher gegenüber sowohl Angriffen von Innen und Außen sein, Daten sicher verwalten können und jederzeit eine konsistente Datenhaltung aufweisen.

Kennzahlen für diesen Bereich könnten beispielsweise die Anzahl der Medienbrüche, Anzahl der Notwendigen Nacharbeitungen, Anzahl der nötigen manuellen Eingriffe oder die Zahl der erfolgreichen Systemangriffe sein.

Zuverlässigkeit

Um die Geschäftstätigkeit dauerhaft zu unterstützen ist die Zuverlässigkeit ein wichtiger Faktor. Das System muss ausgereift genug sein, darf also keine signifikanten Fehler und Lücken mehr aufweisen. Im Fehlerfall muss das System im Stande sein, tolerant zu reagieren und damit die Auswirkungen des Fehlers möglichst klein zu halten. Auch die Wiederherstellung nach einem Ausfall muss schnell und kostengünstig möglich sein.

Messen lassen sich hier die Zahl der bekannten Bugs und notwendiger Patches, die Häufigkeit und Auswirkungen von Systemfehlern und die Zeit, die benötigt wird, um ein System oder Teile nach einem Ausfall wieder herzustellen.

Benutzbarkeit/ Usability

Dieses Thema ist schwer auf Gesamtarchitekturen übertragbar, lässt sich jedoch auf bestimmte Anwendungen innerhalb des Systems anwenden. Diese Anwendungen müssen einfach verständlich sein, die benötigten Funktionen einfach zu finden sein und eine gewisse ästhetisch gestaltete Oberfläche besitzen.

Zur Messung von Usability existieren bereits zahlreiche Verfahren im Softwareentwicklungsbereich, wie z.B. Fragen an Benutzer, Beobachtungen, Eye Tracking. Diese Messergebnisse können auch in die Bewertung der Unternehmensarchitektur mit einfließen.

Effizienz

Zum Bereich Effizienz gehört der schonende Umgang mit Ressourcen (Rechenleistung, Energie, Bandbreite) und eine angemessene Bearbeitungszeit. Die Performance zählt somit auch zu diesem Bereich. Können Nutzeranfragen nicht in einem gewissen Zeitrahmen beantwortet werden, so sinkt der Geschäftsnutzen erheblich.

Zu möglichen Kennzahlen zählen Energieverbrauch, Auslastung der Recheneinheiten oder Kosteneffizienz im Betrieb.

Wartbarkeit

Zur Wartung einer IT-Infrastruktur gehört die Möglichkeit, Probleme analysieren zu können. Dazu bedarf es beispielsweise einer ausführlichen Dokumentation der Prozesse und Funktionen eines Systems. Daraufhin muss es möglich sein, die Struktur und die betroffenen Systeme zu verändern. Diese Veränderung muss so kostengünstig und einfach wie möglich durchzuführen sein, ohne dass der laufende Betrieb dadurch beeinflusst wird. Um die Stabilität der Infrastruktur zu gewährleisten müssen Methoden und Prozesse zum Test von Änderungen und Erweiterungen implementiert werden.

Messbar gemacht werden kann die Wartbarkeit z. B. durch die Anzahl der dokumentierten Funktionen und Prozessmodellen. Das Verhältnis von Test-Dateien zu tatsächlichen Programmfiles gibt Aufschluss über das Ausmaß der durchgeführten Software-Tests.

Flexibilität

Um dem übergreifenden Anspruch der Agilität gerecht zu werden, muss die gesamte Architektur möglichst flexibel gestaltet sein. Der Lebenszyklus von Anwendungen muss durchgängig unterstützt werden, sodass bei Ablösungen bestehende Systeme migriert werden können und unter Umständen eine gewisse Zeit lang nebeneinander koexistieren können. Im Rahmen des Change Management müssen Prozesse für die Migration von Systemen implementiert werden.

Zur Messung der Flexibilität sei auf die in Kapitel 4.1.1 erwähnten Merkmale verwiesen.

4.2.2 Nutzerbezogene Aspekte

Die Qualität aus Nutzersicht spiegelt sich in der Fähigkeit des Systems wieder, dessen Anforderungen zu erfüllen und es ihm zu ermöglichen, seine Ziele effektiv, sicher und zufriedenstellend zu erreichen. Im ISO/IEC-Standard 9126 (ISO 2001) sind die Merkmale *Effektivität*, *Produktivität*, *Sicherheit* und *Zufriedenheit* erwähnt. (vgl. auch Balzert 2008).

Unter Effektivität wird die Fähigkeit des Produktes verstanden, den Benutzer bei der Erreichung seiner Ziele zu unterstützen. Auf Unternehmensarchitekturebene kann darunter ein optimales Alignment von IT und Business verstanden werden, also die Unterstützung der Geschäftstätigkeit durch IT-Systeme. Aus dieser Perspektive lassen sich auch die weiteren Merkmale betrachten: Produktivität beschreibt eine ressourcenschonende Vorgehensweise, Sicherheit die Zuverlässigkeit der Systeme und Zufriedenheit die Adäquatheit der Unterstützung.

Betrachtet man das gesamte Business als Nutzer einer IT-Architektur, so lassen sich dessen Anforderungen unter die nutzerbezogenen Aspekte eingliedern. Aus den Qualitätsmerkmalen für Softwaresysteme (vgl. Bass/Clements/Kazmann 2003, S. 95) lassen sich einige Merkmale auf Unternehmensarchitekturen übertragen. Die Autoren zählen die Merkmale *Time to Market*, *Cost and Benefit*, *Lifetime of the System*, *Target Market*, *Rollout Schedule* und *Integration with Legacy Systems* zu den Business-Qualitäten eines

Softwaresystems. Die für Unternehmensarchitekturen relevanten Komplexe werden nachfolgend beschrieben.

Time to Market

Die Time to Market ist sowohl beim Aufbau einer Unternehmensarchitektur als auch beim regulären Geschäftsbetrieb von Bedeutung. Beim Aufbau ist ein minimaler Zeitaufwand anzustreben, um einerseits Projektkosten zu sparen und andererseits so schnell wie möglich von den Vorteilen profitieren zu können. Während des regulären Betriebs und damit notwendigen Anpassungen wiederholt sich dieser Anspruch ständig. Aus Nutzersicht ist also ein System, das schnell aufgebaut werden kann und die Möglichkeit bietet, schnelle Anpassungen durchzuführen, qualitativ hochwertig in Bezug auf die Time to Market.

Cost and Benefit (Kosten und Nutzen)

Jedes Entwicklungsprojekt hat ein Budget, das nicht überschritten werden sollte. Dabei haben verschiedene Ansätze unterschiedliche Entwicklungskosten. Werden neue Technologien eingesetzt, die im Unternehmen noch nicht verwendet wurden, so sind die Kosten höher, als wenn auf bereits existierende Technologien aufgebaut wird. Allerdings können durch den Einsatz neuer Technologien auf der Nutzenseite Vorteile entstehen. Eine sehr flexible Architektur ist dabei im Regelfalle deutlich teurer als eine weniger flexible, spart dabei im weiteren Lebenszyklus allerdings Kosten bei der Wartung und besonders der Erweiterung und Anpassung.

Lifetime of the System (Lebensdauer des Systems)

Bei Systemen mit einer erwarteten langen Lebensdauer (wie es bei einer Unternehmensarchitektur der Fall ist) werden Aspekte wie Änderbarkeit, Skalierbarkeit und Erweiterbarkeit wichtig. Die Vorbereitung auf diese Ansprüche verlängert jedoch die Time to Market und verursacht höhere Kosten, erweitert gleichzeitig aber die Lebensdauer der Architektur.

Integration with Legacy Systems (Integration bestehender Systeme)

Im Unternehmensumfeld bestehen viele mit der Zeit gewachsene Systeme, die in neue Architekturen eingebettet werden müssen. Dazu ist es notwendig, eine Strategie für die Integration dieser Systeme zu definieren und Schnittstellen zur Anbindung zu definieren. Der Grad der möglichen Übernahme bestehender Systeme beeinflusst dabei sowohl die Entwicklungszeit, da keine Neuentwicklungen erforderlich sind, als auch die Entwicklungskosten. Allerdings muss berücksichtigt werden, in wie fern Altsysteme Lasten darstellen, welche die zukünftige Flexibilität der Architektur negativ beeinflussen können.

4.2.3 Prozessbezogene Aspekte

Im prozessbezogenen Ansatz der Qualitätsbeurteilung wird von der Qualität des Entwicklungsprozesses auf die Qualität des daraus resultierenden Endproduktes geschlossen. Beim Aufbau einer Unternehmensarchitektur kann auf den Entwicklungsprozess

zwar Wert gelegt werden, messbar sind die darin enthaltenen Qualitätsmaßgrößen jedoch nur schwer. Nach van den Bent (2006, S. 12) besteht der Prozess von Unternehmensarchitekturen aus Aktivitäten im Bereich Entwicklung, Wartung, Deployment und Governance einer Unternehmensarchitektur. Bei sämtlichen diesbezüglichen Aktivitäten besteht die Möglichkeit der Qualitätssicherung.

5 Metriken für Unternehmensarchitekturen

In diesem Kapitel werden die für die technische Qualität relevanten Metriken einer Enterprise Architektur dargestellt.

5.1 Einleitung

Das Ziel der Verwendung von Metriken ist es, eine Struktur messbar zu machen. Nachdem sie messbar ist, ist sie auch klassifizierbar, bewertbar und vergleichbar.

„Dies ist eine gute Architektur“ ist eine reine subjektive Aussage und führt auf keinen Fall zu einem sinnvollen Ergebnis im Rahmen der Diskussion über die Qualität einer Enterprise Architektur (vgl. Masak 2005).

Der Physiker Lord Kelvin hat folgenden Satz gesagt:

“When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot express it in satisfactory kind; it may be the beginning of knowledge, but you have scarcely, in your thoughts, advanced to the stage of science.”

(PLA, vol. 1, "Electrical Units of Measurement", 1883-05-03)

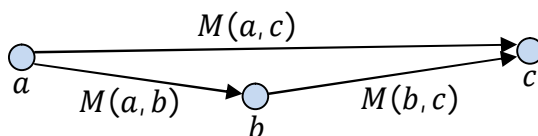
Es gibt direkte und indirekte Messgrößen, die sowohl zur genaueren Bewertung als auch zum Benchmarking von Systemen dienen. Neben solchen Messgrößen existieren auch Größen, die man als Kennzahlen bezeichnet, z.B. der Return on Investment (ROI). Auch andere Größen wie z.B. die Entropie lassen sich im Sinne der Systemevolution bestimmen (vgl. Kapitel 5.6).

5.2 Definition einer Metrik

Die mathematische Definition einer Metrik ist der Abstand zweier Punkte in einem Raum, der folgende Eigenschaften hat:

$$\begin{aligned}
 M(a, b) &\geq 0 & \forall a, b \\
 M(a, a) &= 0 & \forall a \\
 M(a, c) &\leq M(a, b) + M(b, c) & \forall a, b, c
 \end{aligned} \tag{1}$$

Hierbei steht M für die Metrik, a und b sind zwei beliebige Punkte in dem ausgewählten Raum. Die Metrik von a und b ist größer gleich Null, die Metrik von zwei gleichen Punkten entspricht Null. Die Metrik von a und c ist kleiner gleich der Summe der Metrik von a und b und der Metrik von b und c .



Für Softwaremetriken wird zur Vereinfachung als Referenzvektor der Nullvektor gewählt. Jetzt sieht (1) folgendermaßen aus:

$$\begin{aligned} M(a) &> 0 \quad \forall a \neq 0 \\ M(a, b) &\leq M(a) + M(b) \quad \forall a, b \end{aligned} \quad (2)$$

Jede Metrik muss in obigen Gleichungen gegebenen Eigenschaften genügen.

Metriken haben eine sehr starke Aussagekraft, wenn die Messgrößen von einer abstrakten Dimension auf eine quantitative Dimension transformiert werden müssen.

5.3 Graphen

Definitionen der Art der Messung und des Messobjekts sind Voraussetzungen für die Bezifferung der Messgrößen.

Die Enterprise Architekturen werden mithilfe von Graphen repräsentiert. Ein Graph besteht aus einer Menge von Punkten, zwischen denen Linien verlaufen. Die Punkte heißen auch Knoten, die Linien werden auch als Kanten bezeichnet. Die Kanten vertreten die Service-Aufrufe an die jeweiligen Knoten. Das ist normalerweise ein Modell einer Service Oriented Architecture aber es lässt sich auch auf andere Enterprise Architekturen als Modellbildungselement übertragen.

5.3.1 Senke

Ein Knoten, der nur eingehende Kanten hat und keine ausgehenden Kanten, ist eine Senke. Beispiel: Wenn ein Service-Aufruf Daten in das Dateisystem schreibt, ist das Dateisystem eine Senke.

5.3.2 Quelle

Ein Knoten, der nur ausgehende Kanten hat und keine eingehenden Kanten, ist eine Quelle. Beispiel: Ein Messsensor an einem Gerät sendet nur Daten empfängt aber keine.

5.3.3 Gewicht

Das Gewicht ist der Wert einer Kante. In einer Enterprise Architektur hat jede Kante K_{ij} ein Gewicht $w_{ij}(S_l)$, das die mittlere Anzahl der Aufrufe für diesen Knoten pro Service S_l darstellt.

G_{ij} ist die Menge aller Services, die der Knoten A_i vom Knoten A_j nutzen kann.

Weil normalerweise auf dem Niveau einer Enterprise Architektur sehr wenig über die einzelnen Aufrufe und ihre Frequenz bekannt ist, reicht es aus, nur die mittleren Werte zu betrachten.

$$G_{ij} = \sum_l S_l(K_{ij}) \quad (3)$$

Alternativ kann die Wahrscheinlichkeit definiert werden, die wiederum zeigt, wie wahrscheinlich eine Kante benutzt wird.

$$p_{ij}(S_l) = \frac{w_{ij}(S_l)}{\sum_k w_{kj}(S_l)} \quad (4)$$

Die mittlere Service-Wahrscheinlichkeit lässt sich berechnen, indem der Wert in (4) durch die Menge der möglichen Services S_l geteilt wird.

$$\pi_{ij} = \frac{\sum_l p_{ij}(S_l)}{\sum_l S_l} \quad (5)$$

Analog zu (5) kann das mittlere Gewicht w_{ij} wie folgt definiert werden:

$$w_{ij} = \frac{\sum_l w_{ij}(S_l)}{\sum_l S_l} \quad (6)$$

5.4 Arbeitslast

Mithilfe der mittleren Service-Wahrscheinlichkeit können wir die Aufruftrate wie folgt definieren:

$$\delta_i = \begin{cases} f_i & : i \in \text{Senke} \\ \sum_{j=0}^{i-1} \pi_{ij} \delta_j & : \text{sonst} \end{cases} \quad (7)$$

Die Frequenz f_i zeigt den Aufruf einer Senke pro Zeiteinheit. Wenn der Knoten keine Senke ist, d.h. ein anderer Servicepunkt aufgerufen wird, ist die Aufruftrate die Summe aller ausgehenden Aufrufe unter Berücksichtigung der mittleren Service-Wahrscheinlichkeit.

Die Arbeitslast entspricht die Aufruftrate mal Datenmenge pro Aufruf.

$$P_i = \delta_i n \text{ (Bytes)}$$

In einer Enterprise Architektur reicht es, die Arbeitslast identisch zur Aufruftrate zu setzen (Mit der Annahme, dass die Datenmengen gleich groß sind. Datawarehouse-Anwendungen stellen eine Ausnahme dar, da die Datenmengen sehr stark variieren).

5.5 Zeiten und Latenz

Bei einer Enterprise Architektur gibt es primär drei Größen:

- Antwortzeit
- Verarbeitungszeit
- Auslastung (Utilität)

Die Verarbeitungszeit kann wie folgt dargestellt werden:

$$t_i^{verarbeitung} = \begin{cases} \tau_i & : i \in \text{Senke} \\ \tau_i + \sum_{j=0}^{i-1} \pi_{ij} t_j^{Antwort} & : \text{sonst} \end{cases} \quad (8)$$

Dabei ist die innere Service-Zeit τ_i die Zeit, welche der Knoten A_i benötigt um den Aufruf intern abzuarbeiten. Hier wird nur die interne Zeit gewertet. Das Warten auf andere Knoten zählt nicht zur inneren Service-Zeit. Laut der Definition der Senke ist $\tau = t^{verarbeitung}$. Wenn der Knoten keine Senke ist, d.h. ein anderer Servicepunkt aufgerufen wird, ist die Verarbeitungszeit die Summe aller Antwortzeiten unter Berücksichtigung der mittleren Service-Wahrscheinlichkeit.

Anhand der Verarbeitungszeit und der Aufruftrate δ kann die Auslastung eines Knotens A_i wie folgt definiert werden:

$$u_i = \delta_i t_i \quad (9)$$

Die Antwortzeit $t^{Antwort}$ wird durch eine Funktion definiert, die abhängig von der Verarbeitungszeit und der Aufruftrate ist.

$$t_i^{Antwort} = F(\delta_i, t_i^{verarbeitung}) \quad (10)$$

Im Fall einer *M/M/1-Queue* ist die Funktion folgendermaßen definiert:

$$F(\delta, t) = \frac{t}{1 - \delta t} \quad (11)$$

M/M/1-Queue stammt aus der Queuing Theorie. Das erste *M* symbolisiert ankommende zufällige Ereignisse, das zweite *M* eine zufällige exponentiell verteilte Service-Zeit. *1* symbolisiert eine Ressource, die über die Queue zugänglich gemacht wird.

Die entstehende Latenz der Utilität (Auslastung der Ressource) ist für $u \rightarrow 1$ eine sehr stark steigende Funktion (s. Abb. 5.1).

$$\lim_{u \rightarrow 1} F(\delta, t) = \infty \quad (12)$$

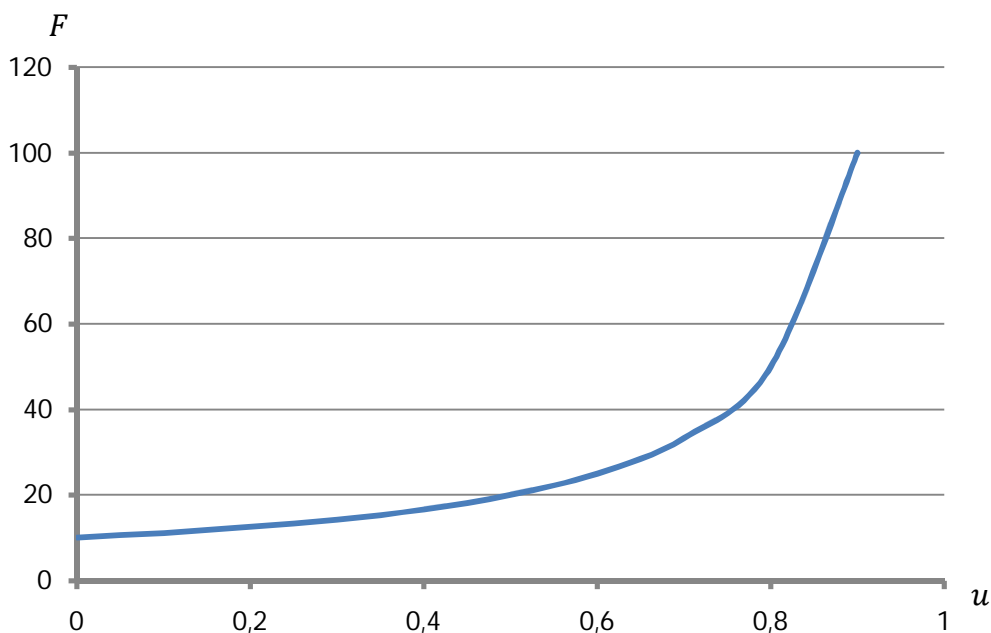


Abbildung 5.1 Die Latenz Funktion (F)

Bei einer Auslastung von 90% ist die Latenz schon um den Faktor 10 gestiegen. Für andere Queue-Mechanismen als $M/M/1$ -Queue sind die Ergebnisse komplizierter. Bei Enterprise Architekturen reichen laut Masak obige Näherung vollkommen aus.

In den meisten Unternehmen sind zwei Formen von Applikationen zu finden:

- Datenintensive Applikationen
- Kommunikationsintensive Applikationen

Bei den datenintensiven Systemen ist die $t^{\text{Antwortzeit}}$ faktisch durch die Datenzugriffszeit gegeben:

$$t^{\text{Antwort Client}} \approx t^{\text{Antwort Database}} \quad (13)$$

Bei den kommunikationsintensiven Applikationen ist die Antwortzeit durch die Kommunikationszeit mit der Latenz determiniert:

$$t^{\text{Antwort Client}} \approx t_{\text{Kommunikation}} + t_{\text{Latenz}} \quad (14)$$

Hier kann man die Service-Zeit τ ignorieren, da sie im Vergleich zu den anderen Zeiten sehr klein ist.

Die Latenz ist als die Zeit definiert, die verstreicht, bis die erste Reaktion eines Systems vorhanden ist. Sie ist in der regulären Antwortzeit enthalten aber nicht in der Service-Zeit, weil diese ohne die Latenz berechnet wird.

Latenz wird primär durch die Zeiten für Instanziierung von Objekten oder dem Aufbau von Verbindungen bestimmt. Für die Graphen kann Latenz sich sehr gut durch

$$t^{Latenz} \approx \sum_{\text{Pfad zur ersten Senke}} t_i^{Latenz} \quad (15)$$

nähern.

Für datenintensive Applikationen ist die Latenz oft durch den Zeitbedarf der ersten Datenbankzugriffe gegeben. Dies entspricht genau der Latenzzeit einer Senke.

5.6 Entropie

Die Entropie stellt ein Maß für eine Unordnung in einem System dar. Die Definition der Entropie von der statischen Mechanik kann auf die Informationstheorie übertragen werden.

Die Entropie eines Graphen ist definiert als:

$$S = - \sum_{j=1}^N P_j \log_2 P_j \quad (16)$$

Hierbei ist P_j die Wahrscheinlichkeit, dass ein Knoten zur Klasse j gehört. Diese Klassifikation kann als die Zahl der einkommenden und ausgehenden Kanten eines Knotens ausgewählt werden.

Wenn alle Knoten gleich sind, ist die Wahrscheinlichkeit $P_j = 1$ und Entropie $S = 0$.

Wenn alle Knoten unterschiedlich sind ist die Wahrscheinlichkeit:

$$P_j = \frac{1}{N}$$

Daraus resultiert die Entropie von:

$$\begin{aligned} S &= - \sum_{j=1}^N \frac{1}{N} \log_2 \frac{1}{N} \\ &= \log_2 N \end{aligned}$$

Für jedes System gilt dann:

$$0 \leq S \leq \log_2 N$$

Mit einem Beispiel kann man die Berechnung der Entropie besser verstehen:

In einem System mit 8 Knoten $G = \{a, b, c, d, e, f, g, h\}$ gibt es folgende Klassifizierung (anhand der Zahl der hinführenden und ausgehenden Kanten)

$$K_1 = \{a, c, f, g\}, K_2 = \{b, d, e\}, K_3 = \{h\}$$

Diese obengenannten Kategorien führen mit ihren jeweiligen Wahrscheinlichkeiten zu der folgenden Entropie:

$$S = -\left(\frac{4}{8} \log_2 \frac{4}{8} + \frac{3}{8} \log_2 \frac{3}{8} + \frac{1}{8} \log_2 \frac{1}{8}\right)$$

$$S = 1.41$$

Die Gesamtentropie eines Systems enthält auch die innere Entropie der Knoten:

$$S_{Gesamt} = S_{Graph} + \sum S_{innere} \quad (17)$$

Hierbei ist die innere Entropie, die Entropie eines Subsystems, das ein Teil des gesamten Systems ist.

Im Rahmen der Enterprise Architekturen ist die Graphenentropie ausschlaggebend. Für die Betrachtung einer Entropie ist es erforderlich beliebige Zwangsbedingungen zu formulieren. Die sinnvollste Zwangsbedingung ist, die Anzahl der Applikationen konstant zu halten:

$$S = S_{Graph} + VS_0 \quad (18)$$

Hierbei ist der Term S_0 die mittlere Entropie pro Applikation und V (die Anzahl der Applikationen) eine Konstante.

5.7 Skalierbarkeit

Die Skalierbarkeit eines Systems ist meistens die Anzahl von Anfragen, die der Antwort-Prozessor innerhalb einer gegebenen Zeit abarbeiten kann, wenn sich die mittlere Antwortzeit linear zu der Zahl der Anfragen verhält:

$$\bar{t} \sim n_{Requests}$$

Wenn ein System skaliert, gibt es zwei Möglichkeiten bei einer wachsenden Anzahl von Anfragen zu reagieren:

- Ein schnellerer Server
- Mehrere Server

Die Skalierungsfunktion verknüpft die Kapazität eines Systems mit einer anderen Größe im gesamten System. Die Kapazität eines Systems kann in drei verschiedene Kategorien eingeteilt werden:

- Informationskapazität

- Verbindungskapazität
- Prozesskapazität

Die Prozesskapazität ist für Enterprise Architekturen von großer Bedeutung. Die Zahl ist als die Anzahl der gleichzeitig in einer gegebenen Zeiteinheit abzuarbeitenden Prozesse zu definieren. Die Verbindungskapazität entspricht der Anzahl der Clients. Die Informationskapazität hat heutzutage relativ wenig Bedeutung, da Speicherplatz preisgünstig ist.

Kapazität ist eine Funktion der Anzahl der Systeme:

$$Capacity = F(n_{System})$$

Im engeren Sinne wird die Skalierbarkeit durch eine lineare Funktion von n_{System} definiert und in der Praxis stellt diese Funktion eine obere Grenze für die tatsächliche Kapazität dar (s. Abb 5.2).

$$Capacity = c_0 + c_1 n_{System}$$

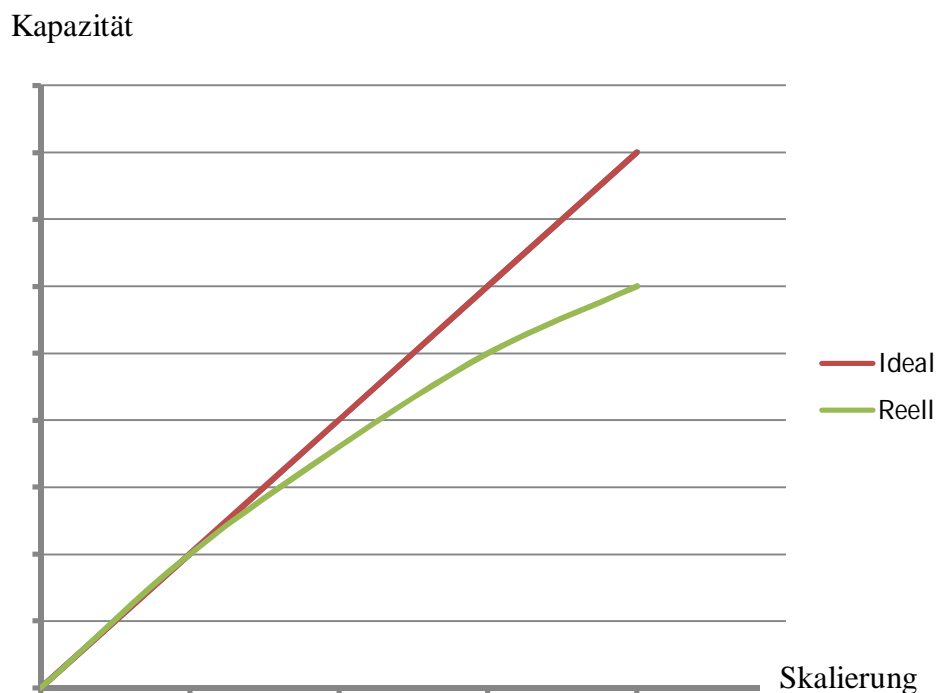


Abbildung 5.2 Die Darstellung von der Kapazität im idealen und realen Fall.

Der Zusammenhang zwischen Koppelung und Skalierbarkeit ist invers proportional (s. Abb. 5.3). Je schwächer die Koppelung, desto höher ist die Skalierbarkeit. Je stärker die Koppelung, desto geringer ist die Skalierbarkeit.

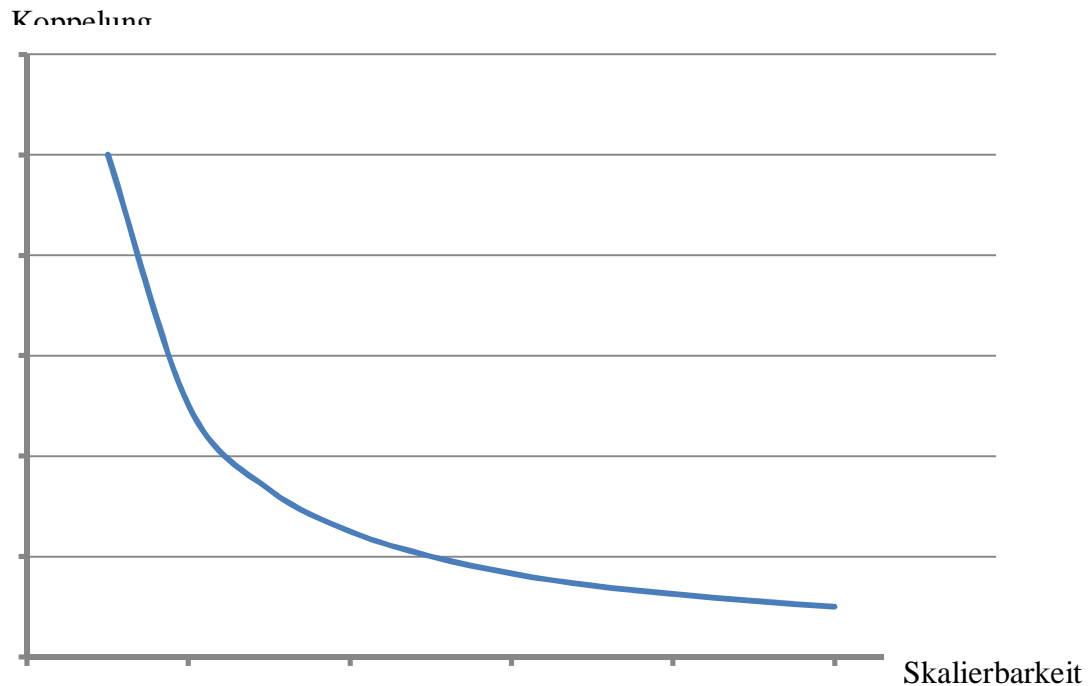


Abbildung 5.3 Die inverse Proportion zwischen Koppelung und Skalierbarkeit

5.8 Zusammenfassung

Die Messung verschiedener Aspekte eines EA Produkts in der Testphase ist erforderlich, da ein Unternehmen im Vorfeld entsprechend reagieren kann. So können potenzielle Risiken beseitigt werden oder verhindert werden, dass ein qualitativ minderwertiges Produkt das laufende Geschäft negativ beeinflusst.

Die Definition einer quantitativen Methode zur Qualitätsmessung durch die Zerlegung eines EA Systems in seine Einzelteile stellt eine große Herausforderung dar.

Für ein Unternehmen ist diese Vorgehensweise aufwändiger als andere abstrakte Methoden (z.B. Balanced Scorecard) anzuwenden oder High-Level-Qualitätsattribute zu definieren. Die Aggregation der gemessenen Eigenschaften und Attribute auf Gesamtsystemebene ist zwar sehr aufwändig, liefert jedoch präzise und zuverlässige Werte.

Eine Enterprise Architektur unter den im diesen Kapitel definierten Aspekten zu evaluieren, ist für eine akkurate Beurteilung der gesamten Qualität unverzichtbar.

6 Zusammenfassung und Ausblick

Angesichts der Bedeutung des Themas Unternehmensarchitekturen ist es bemerkenswert, dass noch kein Standardwerk zur Bewertung der Qualität existiert. Dies könnte eine Folge der Komplexität des Themas sein, oder aber auch in der Variabilität des Begriffes begründet liegen. Erschwerend kommt hinzu, dass eine Architektur immer firmenspezifisch entwickelt werden muss und daher die Standardisierungsmöglichkeiten eingeschränkt sind.

Die Definition von Kennzahlen und Anwendung von Kennzahlensystemen wie der Balanced Scorecard ermöglicht eine umfassende Bewertung der Unternehmensarchitektur. Mittels Metriken lassen sich Kennzahlen auf mathematische Weise erheben, wie in dieser Arbeit dargestellt wurde. Dieser Ansatz erlaubt eine zuverlässige und genaue Messung, was auch zuverlässige Messergebnisse zur Folge hat.

Die Übertragung von Kennzahlensystemen und Messgrößen aus verwandten Bereichen wie der Informationstheorie ermöglicht die Entwicklung eines spezifischen Messinstruments, welches auf bereits bewährten Methoden fundiert.

Weiterer Forschungsbedarf besteht in der Definition und konkreter Entwicklung eines Messinstruments zur Qualitätsbewertung. Dazu gehören beispielsweise die Definition von Kennzahlen aus unterschiedlichen Bereichen, die Anpassung des Bewertungssystems und die Untersuchung der die Qualität beeinflussenden Faktoren.

Literaturverzeichnis

- Achert, W. (2005): IT-Qualität im Unternehmen. In: der EDV-Leiter 01/05.
- Aier, S., Dogan, T. (2005): Indikatoren zur Bewertung der Nachhaltigkeit von Unternehmensarchitekturen. In: Ferstl et al. 2005.
- Balzert, H. (2008): Lehrbuch der Softwaretechnik: Softwaremanagement. 2. Auflage. Heidelberg: Spektrum Akademischer Verlag.
- Bass, L., Clements, P., Kazman, R. (2003): Software Architecture in Practice. 2. Aufl., Boston: Addison-Wesley.
- Dove, R., Hartman, S., Benson, S. (1996): An Enterprise Reference Model. Agility Forum, Report AR96-04. <http://www.parshift.com/docs/aermodA0.htm>
- Ferstl, O., Sinz, E., Eckert, S., Isselhorst, T. (Hrsg.) (2005): Wirtschaftsinformatik 2005: eEconomy, eGovernment, eSociety. Heidelberg: Physica-Verlag.
- Garvin, D. A. (1984): What does „Product Quality“ Really Mean?. In: Sloan Management Review Fall 1984, S. 25-43.
- Hagen, C., Schwinn, A. (2006): Measured Integration. In: Schelp/Winter 2006.
- ISO (2001): ISO/IEC 9126-1 - Software engineering - Product quality - Part 1: Quality model. International Organization for Standardization/International Electrotechnical Commission.
- Malich, S. (2008): Qualität von Softwaresystemen. Wiesbaden: Gabler Verlag.
- Masak, D. (2005): Moderne Enterprise Architekturen. Berlin Heidelberg: Springer-Verlag.
- Schelp, J., Winter, R. (Hrsg.)(2006): Integrationsmanagement. Berlin Heidelberg: Springer-Verlag.
- Schwinn, A., Winter, R. (2005): Entwicklung von Zielen und Messgrößen zur Steuerung der Applikationsintegration. In: Ferstl et al. 2005.
- Van den Bent, B. (2006): A Quality Instrument for the Enterprise Architecture Development Process. Master's thesis, Utrecht University & Sogeti Netherlands B.V.
- Vom Brocke, J., Sonnenberg, C., Thurnher, B., Müller, B. (2008): Wertorientierte Gestaltung von Unternehmensarchitekturen. In HMD – Praxis der Wirtschaftsinformatik, Heft 262, August 2008. Heidelberg: dpunkt.verlag.
- Winter, R., (2006): Ein Modell zur Visualisierung der Anwendungslandschaft. In: Schelp/Winter 2006.