

Enterprise Architecture als Katalysator zwischen Qualität, Effizienz und Governance

Andreas Hartmann¹

Abstract: Die Digitalisierung treibt Änderungen in Unternehmen unnachlässig an und Geschäftsprozesse ohne entsprechende Softwareunterstützung werden selten. Um hier erfolgreich zu sein müssen Entscheidungsprozesse frühzeitig in der Softwarearchitektur abgebildet werden. Sie müssen der Unternehmensstrategie folgen und die Entscheidungsstrukturen des Unternehmens abbilden. Enterprise Architecture modelliert diese Informationen. Somit muss Enterprise Architecture eine Rolle als Bindeglied zwischen Strategie und Operative, z.B. als Architekturmuster in der Softwareentwicklung, einnehmen. Das erhöht nicht nur die Qualität der eingesetzten Software sondern macht Änderungen beherrschbar und steuerbar.

Keywords: Enterprise Architecture, IT-Governance, Software Architecture, DevOps

1 Einleitung

Aktuell nehmen sowohl die Größe, die Funktionalität und die gegenseitigen Abhängigkeiten von IT-Landschaften zu – die Komplexität steigt an [Ha16]. Experten meinen, dass nur durch einen modellbasierten Ansatz in Kombination mit einem modularen Aufbau diese Landschaften in Zukunft noch zu steuern sind [Sc17]. Gleichzeitig ist die Digitalisierung noch lange nicht abgeschlossen. Besonders deutlich werden Unterschiede, wenn Aspekte der Enterprise Architecture (EA) [RZM14] mit Softwarearchitekturen und Entwicklungsparadigmen verglichen werden [So11]. Als Beispiel können der Begriff der Governance und die damit verbundenen Entscheidungsprozesse herausgegriffen werden. Der vorliegende Beitrag beschreibt eine Untersuchung, wie Software - *by Design* - zu einer Erhöhung der Effizienz und Effektivität von Entscheidungsprozessen beitragen kann. Die EA nimmt herbei eine Rolle als Bindeglied/Katalysator ein.

2 Grundlagen

Software kann nur so gut sein wie sie zusammen mit Menschen funktioniert. Menschen, die in einer agilen und digitalisierten Welt sowohl effektiv als auch effizient über Situationen und Fragen entscheiden müssen. Eine Entscheidung kann nicht optimal getroffen werden wenn Informationen fehlen, der Kontext der Entscheidung nicht klar ist oder sich der Entscheider erst umständlich in einer Software zurechtfinden muss (vgl. [Is15]). Mit

¹ Hochschule für Telekommunikation Leipzig, Praktische Informatik/Software Engineering, Gustav-Freytag-Str. 43-45, 04277 Leipzig, hartmann@hft-leipzig.de

der Komplexität von IT-Landschaften steigen die Herausforderungen. Die Digitalisierung ist offensichtlich eben dort unvollständig gedacht, wenn Informationen aus einer Software erst extrahiert werden müssen, um dann manuell zu einem Termin gebracht zu werden, wo schließlich eine benötigte Entscheidung getroffen werden kann.

2.1 Softwareentwicklung und Softwarearchitektur

Werkzeuge hinsichtlich der Informationsbereitstellung sind bekannt und im Einsatz [Ha16-1]. Die Entscheidung selbst wird heute allerdings oft noch außerhalb dieser Werkzeuge getroffen. Vollständig betrachtet müsste die Entscheidung jedoch wieder den Weg zurück finden und der automatisierte Prozess direkt weiterlaufen. [Br14] zeigt z.B., wie ein ganzheitlicher Ansatz im ERP Kontext auszusehen hat. Im Sinne einer fachlichen Trennung der Belange stellt sich die Frage, ob die entsprechende Funktionalität der einzelnen Software nicht gekapselt und als Schnittstelle oder Service bereitgestellt werden sollte. Vergleichbar mit einem Prozess der Anmeldung könnte der Entscheidungsbedarf weitergeleitet, an entsprechender Stelle entschieden und dann die Entscheidung wieder in das System zurückgespielt werden (vgl. [Wi12], [Fo11]). Viele Entwicklungsmethoden/-modelle sind hier immer noch zu zentriert auf das eigene IT-System ausgelegt und Schnittstellen eher für den reinen Datenaustausch vorbehalten. Dabei enthält eine strukturierte Anforderungsanalyse alle Informationen zu Stakeholdern und Entscheidungsbedarfen. (Agile) Projektvorgehensmodelle sorgen für die ausreichende Dokumentation. Anstelle des üblichen Nutzerdialogs könnten der Entscheidungskontext und eine Entscheidungsvorlage gekapselt und als Service gestaltet werden.

2.2 Enterprise Architecture

Aus Unternehmenssicht genügt es nicht, sich nur auf einzelne Anwendungen zu fokussieren. Einen ganzheitlichen Ansatz zu finden ist jedoch nicht in jedem Fall eine einfache Frage – insbesondere bei größeren und gewachsenen Organisationen. Enterprise Architecture beschreibt die ebendiese Unternehmenssicht und heute gibt es verifizierbare Referenzarchitekturen. [CD17] zeigen am Beispiel des Telekommunikationssektors eine solche ganzheitliche Lösung mit entsprechenden EA Referenzmodellen.

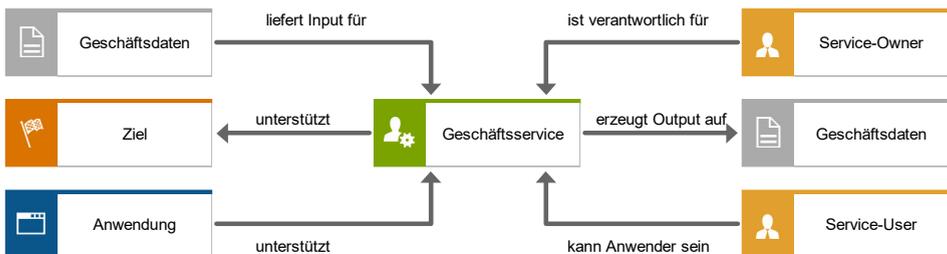


Abb. 1: Geschäftsarchitektur mit Detailbeschreibung eines Services

Die EA ist zudem ein wichtiger Baustein für die IT-Governance und liefert essentielle Informationen zu den Entscheidungsstrukturen [Ni06]. Für den Fall dass die Geschäfts-sicht konsequent modelliert wurde, liegen die Angaben über Zuständigkeiten und Verantwortlichkeiten vor. Über die Zuordnung von Software zu Prozess oder Service können die Entscheider identifiziert und benannt werden. Abb. 1 fokussiert einen Geschäfts-service mit Zuordnung der Verantwortlichkeit (Rolle: *Service-Owner*), Anwender und unterstützende IT (*Anwendung*).

2.3 IT-Governance

Mit IT-Governance verfolgen [WR04] das Ziel: „*Specifying the decision rights and accountability framework to encourage desirable behavior in the use of IT.*“ Werden nach diesem Prinzip alle Informationen zu den Verantwortlichkeiten in der EA modelliert, ergibt sich nicht nur ein Gesamtbild des Unternehmens, sondern vielmehr die Transparenz der Entscheidungsstrukturen. [WR04] definieren darüber hinaus noch zwei unterschiedliche Aspekte hinsichtlich der IT-Governance: wer trifft die Entscheidung und wer bringt die zur Entscheidung benötigten Informationen bei. Ein vollständiges EA-Modell würde daher beide Aspekte implementieren. In Abb. 2 wird der Ausschnitt einer EA-Modellierung betrachtet. Dargestellt sind die Geschäfts-services und deren hierarchische Unterteilung mit je einem einzelnen Ausschnitt je Ebene. Das Gesamtbild ergibt sich, wenn für jeden Geschäfts-service die Verantwortlichkeiten und die zugeordnete IT-Unterstützung modelliert werden – in diesem Fall also für den Geschäftsbereich IT-Infrastruktur Management. Ein Ziel der Orchestrierung ist die Transparenz, welche direkt über ein geeignetes Reporting erreicht werden kann. DevOps-Werkzeuge wie z.B. Puppet liefern die dafür benötigten Fähigkeiten [Ak15].

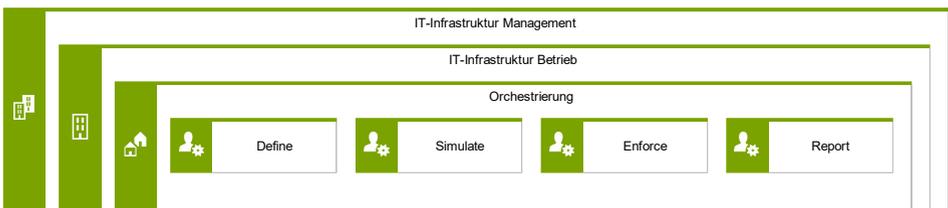


Abb. 2: Servicearchitektur eines Geschäftsbereichs

3 Konzept und Untersuchung

Die Untersuchung geht davon aus, dass Informationen zur Governance z.B. über entsprechende Referenzarchitekturen modelliert sind. Darauf basierend kann jetzt die Ver-

bindung zur Softwareentwicklung hergestellt werden. Beginnend mit einem Muster, nach dem die Kapselung der Entscheidungsfunktion bereits in der Softwarearchitektur verankert wird, können die konkreten EA-Modelle in die Anforderungen aufgenommen und die benötigten Schnittstellen spezifiziert werden. Im Sinne von MicroServices werden die Entscheidungen als funktionale Einheiten separiert. Die Umsetzung kann hier auf zwei Wegen erfolgen. Im ersten Fall findet eine angebotsorientierte Implementierung statt, bei der die Entscheidungsfunktion von der Software in Richtung Entscheider kommuniziert wird. Die tatsächliche Kommunikation könnte sich zunächst auf bekannte Medien stützen.

In einem erweiterten Szenario kommt ein bedarfsorientiertes Framework (*Generic Decision Framework*) zum Einsatz, welches Entscheidungsfunktionen aus der gesamten IT-Landschaft bündelt und die Entscheider nach Bedarf einbindet. Hier können die Entscheidungen zusammen mit den benötigten Informationen konsolidiert verarbeitet werden. Das Framework richtet seine fachlich Struktur dabei offensichtlich am EA-Modell und damit an der Governance aus. Es wird sichergestellt, dass die Entscheidungsbedarfe effektiv und effizient an die Entscheider gereicht werden. Das Framework kann dabei hierarchisch aufgebaut sein – passend zum Geschäftsmodell. Das bedeutet, Entscheidungen können bei Bedarf eskaliert bzw. an eine höhere Instanz weitergereicht werden. Auch eine dezentrale Architektur des Frameworks ist möglich und als Kollaborationswerkzeug unterstützt es die bidirektionale Zusammenarbeit (im Unterschied zum reinen Management Information System).

Im Beispiel der Orchestrierung (DevOps) kommt die Software Puppet zum Einsatz. Neben dem Reporting stehen Aufgaben zur automatischen Steuerung der Infrastruktur an. Entscheidungen bzgl. der Steuerung können mit Hilfe der Software getroffen und anschließend umgesetzt werden, wie z.B. der Umzug oder die Abschaltung von Anwendungscontainern. Die Verantwortung kritischer Entscheidung liegt organisatorisch beim Verantwortlichen der Orchestrierung bzw. wird von dort entsprechend der Servicehierarchie eskaliert. Abb. 3 zeigt schematisch die Kapselung der Entscheidungsfunktion, die Verantwortlichkeit und Verknüpfung zum generischen Framework. Die Informationsstruktur des Frameworks leitet sich aus der Governance ab.

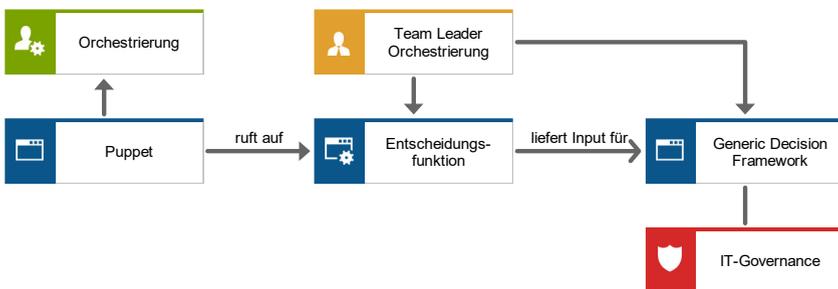


Abb. 3: Erweiterung des IT-Systems Puppet und Verknüpfung mit dem Framework

Das Konzept verbessert jedoch nicht nur die Effizienz der Entscheidungsprozesse. Im Unterschied zu anderen Methoden (vgl. [RR13]) kann der Ansatz auch strukturbildend oder als Qualitätsmaß benutzt werden (ähnlich [CR13], wo der Fokus auf Daten liegt).

Qualitätsmessung („gelebte Strukturen“): Die EA-Modelle müssen nicht der gelebten Governance entsprechen. Oft haben sich Verantwortlichkeiten geändert oder die Modelle sind nicht aktuell. Das generische Framework macht die gesamten Entscheidungsprozesse transparent und prüfbar. Besonders deutlich werden Abweichungen, wenn der Entscheidungsbedarf im Framework eine falsche Rolle erreicht.

Optimierung der Architektur: IT-Landschaften sind nicht statisch sondern entwickeln sich evolutionär weiter. EA-Modelle der Governance und Informationen des generischen Frameworks können in dieser Hinsicht als Gestaltungsprinzip für die IT-Landschaft herangezogen werden. Applikationen werden so geplant, dass möglichst effektive und effiziente Entscheidungen möglich sind. So können z.B. Schwachstellen sichtbar gemacht und in effizientere Strukturen umgewandelt werden.

Verwendung der Schnittstellen für Protokollierung: Schließlich kann mit der Kapselung der Entscheidungsfunktion eine kontinuierliche Protokollierung von Entscheidungen stattfinden. Neben der Transparenz ist hier ein wichtiger Nachweisfaktor für automatisierte Arbeitsvorgänge zu sehen.

4 Diskussion und Ausblick

Die Untersuchung adressiert eine Lücke zwischen Softwareentwicklung und Enterprise Architecture. In Zeiten von Digitalisierung und steigender Komplexität von IT-Landschaften wird Software viel zu oft noch aus zentralisierter Sicht entworfen. Probleme mit Schnittstellen zu anderen Anwendungen oder gar eine Bewertung der Qualität aus Sicht der IT-Landschaft fließen nicht konsequent in die Softwarearchitektur ein. Am Beispiel von Entscheidungsprozessen werden in der Arbeit Softwarestandards untersucht und evaluiert. Auf Basis von entsprechenden EA-Modellen mit Informationen zur IT-Governance wird weiterhin ein generisches Framework entwickelt, welches den Gedanken der Digitalisierung vorantreibt. Dabei wird zugleich ein modernes Architekturmuster erarbeitet, welches sich auch auf andere Bereiche übertragen lässt. Dieses Prinzip eröffnet eine agile Kopplung zwischen Strategie und Operative – so wie es [Ah17] mit dem Begriff *S2E* für die Zukunft von EAM vorhersagen. Gleichwohl ist zu beachten, dass Agilität in der EA eine Herausforderung darstellt. Veraltete Modelle bergen Risiken für die Effektivität des Frameworks. Darüber hinaus bedeutet das Architekturmuster eine „Öffnung“ für die Softwarehersteller, was u.U. ein umfangreiches Re-Design von komplexen Anwendungen zur Folge haben kann. Die Wirtschaftlichkeit wird hierbei sorgfältig zu beobachten sein.

Literaturverzeichnis

- [Ah17] Ahlemann, F., Stettiner, E., Messerschmidt, M., Legner, C. (Eds.): *Strategic Enterprise Architecture Management. Challenges, Best Practices, and Future Developments*. Springer, Heidelberg, 2017.
- [Ak15] Akshaya H. L.; Nisarga Jagadish S.; Vidya J.; Veena K.: *A Basic Introduction to DevOps Tools*. *International Journal of Computer Science & Information Technologies* Volume 6 Issue 3 05-06/15, S. 2349-2353, 2015.
- [Br14] Brockmann, C.: *An approach to design the business model of an ERP vendor*. Gito, Berlin, 2014.
- [CR13] Capirossi J., Rabier P.: *An Enterprise Architecture and Data Quality Framework*. In (Benghozi P.J., Krob D., Rowe F. (eds)): *Digital Enterprise Design and Management 2013. Advances in Intelligent Systems and Computing*, Vol 205. Springer, Berlin, Heidelberg, S. 67-79, 2013.
- [CD17] Czarniecki, C., Dietze, C.: *Reference Architecture for the Telecommunications Industry: Transformation of Strategy, Organization, Processes, Data, and Applications*. Springer, Heidelberg, 2017.
- [Fo11] Fowler, M.: *Patterns of Enterprise Application Architecture*, Addison-Wesley Longman Publishing Co., 2011.
- [Ha16] Hanschke, I.: *Enterprise Architecture Management – einfach und effektiv*, 2. Auflage, Carl Hanser Verlag, 2016.
- [Ha16-1] Hafen, T.: *com! Professional*, <http://www.com-magazin.de/praxis/software/professionelle-datenanalyse-bi-tools-1061106.html>, Stand 14.05.2017.
- [Is15] ISO 9241-210:2010 - *Ergonomics of human-system interaction*, www.iso.org/standard/52075.html, Stand: 14.05.2017.
- [Ni06] Niemann, K. D.: *From enterprise architecture to IT governance*. Vol. 1. Springer Fachmedien, 2006.
- [RR13] Roszkowski J., Roszkowska A.: *The Quality Management Metamodel in the Enterprise Architecture*. In (Skersys T., Butleris R., Butkiene R. (eds)): *Information and Software Technologies. ICIST 2013. Communications in Computer and Information Science*, Vol 403. Springer, Berlin, Heidelberg, S. 11-21, 2013.
- [RZM14] Roth, S.; Zec, M.; Matthes, F.: *Enterprise Architecture Visualization Tool Survey 2014*. Technical Report. sebis, Technische Universität München. 2014.
- [Sc17] Schwaninger, M.: *Komplexität systemisch meistern*. *Wirtschaftsinformatik & Management* 02/17, S. 20-21, 2017.
- [So11] Sommerville, I.: *Software Engineering*, 9th ed., Pearson, 2011.
- [Wi12] Wiggins, A.: *The Twelve Factors*, <https://12factor.net>, Stand: 14.05.2017.
- [WR04] Weill, P.; Ross J. W.: *IT governance: How top performers manage IT decision rights for superior results*. Harvard Business Press, 2004.