

GSem Ausarbeitung: Überblick EAM

Chancen Aufstrebender Technologischer Paradigmen wie Continuous Delivery, Microservices und Mobile First für Modernes Enterprise Architecture Management

Daniel Kirchner

HAW Hamburg, 20099 Hamburg, Germany

Abstract. Es wird eine Auswahl von neuen technologischen Paradigmen betrachtet, bei denen zur Zeit wachsendes Interesse und erste positive Erfahrungen in dem Betrieb und der Entwicklung von Software beobachtet werden. Aus diesen wird versucht einen allgemeinen Trend zu erkennen und es werden Verknüpfungspunkte zu klassischen Frameworks aus dem Enterprise Architecture Management und der IT Governance vorgeschlagen. Die Ergebnisse werden unter den Gesichtspunkten *lean*, *agile* und *collaborative* diskutiert.

1 Motivation

Informationstechnologie hat sich als lebenswichtiger Aspekt in allen Bereichen großer Unternehmen längst etabliert. Selbst in Unternehmen deren Kerngeschäft nicht in der Informationstechnologie selbst liegt werden Geschäftsprozesse - bewusst oder unbewusst - in der Architektur und dem Datenaustausch der betriebenen Systeme abgebildet.

Produkte selbst, ihre Entwicklung und Verkauf, der Kontakt zu Kunden und Lieferanten, Supply-Chain, die Verwaltung der Mitarbeiter und Prozesse aus Controlling und Finanzwesen sind nur wenige Beispiele aus einer sehr langen Liste von Bereichen, die inzwischen von Softwareprodukten mehrerer Generationen unterstützt und teilweise automatisiert werden.

Dabei werden nicht nur Daten erzeugt, verteilt und konsumiert, sondern zunehmend wird auch eine intelligente Verarbeitung der eigenen Daten automatisiert. Dabei werden z.B. Aufgaben aus der Planung, Anomalieerkennung und Vorhersage automatisiert, die vorher von Menschen übernommen wurde. Dabei entstehen wiederum neue Anforderungen an eine systematische Datenerfassung von neuen - und insbesondere auch von bestehenden - Systemen.

Das Problem die IT-Landschaft eines Unternehmens ständig neuen Bedürfnissen anzupassen und dabei einen möglichst schlanken und stabilen Betrieb sicherzustellen ist seit Jahrzehnten Gegenstand von wissenschaftlicher und industrieller Forschung.

In den letzten Jahren haben Techniken aus der Softwareentwicklung mit pragmatischen Ansätzen, kurzen Feedbackzyklen und kleinen Teams gute Resultate in der

Umsetzung komplexer Softwareprojekte erzielt und die Aufmerksamkeit des *Enterprise Architecture Managements* erregt [5].

In diesem Aufsatz werden konkrete Möglichkeiten erläutert aufstrebende technologische Paradigmen in eine *Enterprise Architecture* zu integrieren und Verknüpfungen zum *Enterprise Architecture Management* erläutert.

2 Grundlagen des Enterprise Architecture Managements

2.1 Definition und Scope

Nach [3] ergeben sich folgende Arbeitsdefinitionen von *Enterprise Architecture* und *Enterprise Architecture Management*:

Enterprise Architecture (EA) ist eine Repräsentation der Struktur und des Verhaltens der IT-Landschaft eines Unternehmens mit Bezug auf das geschäftliche Umfeld. Dabei stellt sie die momentane und die zukünftige Nutzung von IT im Unternehmen dar und liefert einen Plan zur Erreichung eines zukünftigen Zustands. Dabei bietet sie

- Einsichten in die IT-Nutzung aus Sicht des Geschäftsbetriebs
- eine Vision für die zukünftige Nutzung von IT im Geschäftsbetrieb
- einen Plan für schrittweise Evolution hin zu einem zukünftigen Zustand

Enterprise Architecture Management (EAM) ist ein strukturierter Ansatz um EA zu erzeugen, zu verwalten und anzuwenden um die IT-Nutzung am Geschäftsbetrieb auszurichten. Dabei übersetzt EAM die geschäftliche Vision in konkrete Unternehmungen und begleitet das Unternehmen vom jeweils aktuellen EA-Zustand bis zu einem Zielzustand.

2.2 Tools und Frameworks

Um einen gemeinsamen Wortschatz der Beteiligten des EAM zu schaffen, aber auch um eine Sammlung von bewährten Richtlinien, Prozessen und Dokumenten verfügbar zu machen wurden verschiedenste EAM-Frameworks entwickelt.

Zu den wichtigsten (siehe [3]) Frameworks gehören das Zachman Framework¹, das Open Group Architecture Framework (TOGAF)² und die Gartner Methodologies³.

¹ <https://www.zachman.com/about-the-zachman-framework>, abgerufen am 03.06.2015

² <https://www.opengroup.org/togaf/>, abgerufen am 03.06.2015

³ <http://www.gartner.com/technology/research/methodologies/methodology.jsp>, abgerufen am 03.06.2015

3 Übersicht der betrachteten Technologischen Paradigmen

3.1 Continuous Delivery

Continuous Delivery ist eine Erweiterung von *Continuous Integration*, die als Methodik der Softwareentwicklung bereits erfolgreich in Unternehmen etabliert ist ([6]).

Wesentliches Ziel ist es schnelle Rückmeldung für eingepflegten Code durch möglichst unmittelbare Auslieferung der Änderung bis in das laufende System zu erreichen. Dabei wird eine Automatisierungslinie (*continuous delivery pipeline*) geschaffen, die Build, Tests und Deployment des geänderten produktiven Systems soweit automatisiert, dass ein möglichst schnell Werte (im Sinne von produktiv eingesetzten Features) geschaffen werden (Vgl [wolff]).

3.2 Microservices

3.3 DevOps

3.4 Mobile First

4 Vorschläge zur Umsetzung der Paradigmen auf Unternehmensebene

5 Mögliche Einflüsse auf EAM

6 Zusammenfassung und Ausblick

Es wurde erläutert, dass aus einer sehr oberflächlichen Betrachtung heraus durchaus die Möglichkeit bestehen könnte durch konkrete Einflechtung der genannten technologischen Paradigmen einen erhöhten Reifegrad (*Maturity Level*) bezüglich Agilität und Kollaboration durch und innerhalb von Unternehmens-IT zu erreichen.

Es stellen sich jedoch viele offene Fragen, die in weiterer Literaturrecherche und auch im Experiment behandelt werden müssten, beispielsweise:

- Wie wird eine sinnvolle Aufteilung in Geschäftsbereiche gemessen?
- Welche Metriken existieren
- Wie groß ist der Einfluss von Duplikation von Komponenten?
- Wie geht das vorgeschlagene Modell mit Korrosion seiner Struktur um?
- Welche Artefakte von üblichen EAM-Frameworks lassen sich noch anwenden, welche müssen angepasst werden?
- Wie könnte eine Transition von einer bestehenden Architektur aussehen?
- Welchen hat internationale Verteilung auf das Modell?
- Welchen Einfluss hat das Modell auf Sicherheits- und Verfügbarkeitsrisiken?

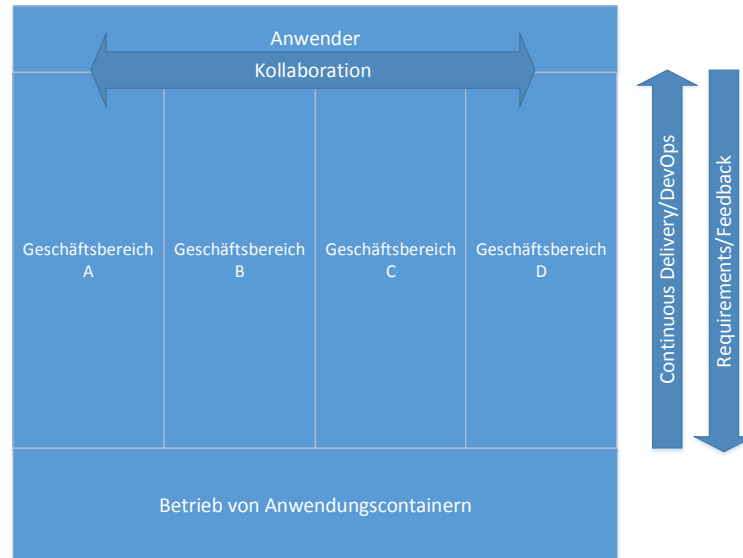


Fig. 1. Naive Sicht auf eine Continuous-Delivery-zentrierte Unternehmensarchitektur

Das *HAW Labor für Anwendungsintegration* ist dabei ein Bereich, der für die Untersuchung vieler dieser Fragen einen akademisch geschlossenen und dennoch realitätsnahen Bereich bietet.

Ein konkreter Einstieg könnte sein, im Rahmen eines Grundprojektes die technischen Voraussetzungen für eine *Continuous Delivery Pipeline* in einem technisch und geschäftlich heterogenen Umfeld zu schaffen.

Durch zweimal jährlich wechselnde Studentengruppen (in der Rolle der Softwareentwickler) und die Abwesenheit des Risikos geschäftsvernichtender katastrophaler Systemausfälle sind dann die Rahmenbedingungen für weiterführende Untersuchungen und Experimente gegeben.

[?] [1]

References

1. Clarke, F., Ekeland, I.: Nonlinear oscillations and boundary-value problems for Hamiltonian systems. *Arch. Rat. Mech. Anal.* 78, 315–333 (1982)
2. Kim, S., Park, S.: Automated Continuous Integration of Component-Based Software: An Industrial Experience ASE '08 Proceedings of the 2008 23rd IEEE/ACM International Conference on Automated Software Engineering Pages 423-426
3. Bente, S.: Collaborative Enterprise Architecture 2011 Morgan Kaufmann Publ.
4. Haynes, S., Skattebo, A.: Collaborative architecture design and evaluation DIS '06 Proceedings of the 6th conference on Designing Interactive systems Pages 219-228
5. Buckl, S., Matthes, F.: Towards an Agile Design of the Enterprise Architecture Management Function 2011 15th IEEE International Enterprise Distributed Object Computing Conference Workshops Pages 322-329
6. Fitzgerald, F., Stol, K.-J.: Continuous software engineering and beyond: trends and challenges RCoSE 2014 Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering Pages 1-9
7. Wolf, D.: Continuous Delivery: Der pragmatische Einstieg 2014 dpunkt.verlag GmbH