Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Vorlesung Objektorientierte
Komponenten-Architekturen
(OOKA)
SS 2022

Prof. Dr. Sascha Alda

9. Mai 2022

Übungen zur Vorlesung
Objektorientierte Komponenten-Architekturen
SS 2022

Übungsblatt Nr. 4
(Abgabe bis: Montag, den 16. Mai 2022, **12:00 Uhr**)

Aufgabe 1 (Case Study für Microservices, FA. WirSchiff*en*Das)

Das Unternehmen WirSchiff*en*Das GmbH & Co KGaA hat über viele Jahre hinweg eine *monolithische* Software-Plattform für den Vertrieb ihrer Produkte (Komponenten für Hochsee-Yachten) entwickelt. Man erkennt, dass Teile der resultierenden Software-Architektur *in die Jahre* gekommen sind. Eine Baustein-Sicht nebst diverser Erläuterungen und Meinungen seitens des Entwicklungsteams zu der Software-Architektur hat man ihnen bereitgestellt, sie finden dieses auf LEA.

Der zuständige Software-Architekt des Entwicklungsteams vertraut ihnen die Aufgabe an, diese Architektur neu zu definieren. Man hat schon über den Architekturstil Microservices gehört, auch über „angrenzende Technologien“ und Methoden, hat aber bisher keine Erfahrungswerte. Man bittet sie, auf Grundlage von Microservices eine neue Software-Architektur zu modellieren.

Darüber hinaus ist man auch über weitere organisatorische Aspekte interessiert, die sich aus dieser Verbesserung der Software-Architektur ergeben. Dazu hat man ein paar Statements aus einem Meeting der Bereichsleitung des Bereichs „Technik“ zusammengetragen, wo vor kurzem der Ansatz Microservices kurz vorgestellt und *leidenschaftlich* diskutiert wurde.

Statement Nr. 1:
„*Wir haben eine strenge Komponenten- und Schichtenbildung, entsprechend haben wir auch Komponenten- und Schichtenteams. Das kann nur Vorteile haben, das wird übernommen*!“! Nach kurzer Diskussion beruht man auch für Microservices auf diese Einteilung und verspricht sich dadurch einen klaren Vorteil. Was denken sie dazu? Was würde sie dem Unternehmen eher empfehlen?

Statement Nr. 2:
„*Das wird schon! Die Geschäftsführung des Unternehmens muss doch für so eine geringfügige Architekturänderung nicht informiert werden*.“ Was denken sie über dieses Statement?

Statement Nr. 3:
„*Wir können doch schon Cloud – wir sind cloud-ready!*“! Da man die Anwendungsschicht auf *einer* Virtuellen Maschine (VM) deployed hat, resultiert man konsequenterweise, dass diese VM auch einfach auf einen Cloud-Anbieter (z.B. Amazon WS) ausgelagert werden kann. Die Anwendung gilt somit als *skalierbar*! Es besteht eh keine Notwendigkeit der Skalierung einzelner Komponenten, so die Bereichsleitung.
Wie stehen sie zu diesem Urteil?

Statement Nr. 4:
„*Ach immer dieses API-Team...*“. Das „API-Team“ (vgl. Baustein-Sicht) hat sich wohl in letzter Zeit oft über eine zunehmende Komplexität der Komponente „GP-API“ aus dem Presentation Logic Layer beschwert, gibt aber Begründungen nicht explizit an (-- man gibt zu, vielleicht auch nicht richtig hingehört zu haben...). Was für Gründe könnten das sein aus ihrer Sicht? Wie könnte man diese Schicht ggf. im Rahmen der Microservice-Architektur optimieren?

Statement Nr. 5:
„*Die Analyse-Komponente ist DAS Problem überhaupt!*“! Leider sorgt die Analyse-Komponente innerhalb der Komponente „Manufacturing Products“ häufig für Probleme während der Konstruktion von Produkten (siehe Erläuterungen in Dokument). Was sind hier konkret die Probleme? Wie sähe eine mögliche Lösungsstrategie aus?

Statement Nr. 6:
„*Unsere langfristige Strategie: ein globales und universales Datenmodell!*“! Da man auch immer noch diverse Bestandsysteme verwendet (SAP ERP, Oracle CRM), sieht man die Etablierung eines gemeinsamen universalen Datenmodells als unabdingbar. Was denken sie? Wie würden sie hier mit den heterogenen Datenmodellen umgehen? Wird diese Meinung von allen Entwicklerteams mitgetragen?

Statement Nr. 7:
„*Das Entwicklungsteam will ja unbedingt ein Monitoring. Warum das bitte?? Zu teuer und wenig aussagekräftig. Das lassen wir mal!“.* Was sind aus ihrer Sicht die Vorteile einer Monitoring-Komponente? Gibt es Herausforderungen?

Ihre Aufgaben:

a.)
Gehen sie auf die o.g. Statements kritisch ein und beantworten sie die aufgeführten Fragen in 2-3 Sätzen. Die resultierenden Erkenntnisse können sie natürlich *später* gerne in die Architektur einfließen lassen. Beachten sie dabei auch die Erläuterungen aus der Architekturbeschreibung.

Zur Diskussion der Vor- und Nachteile von Komponenten-, Schicht- und Funktionsteams empfehle ich den Artikel von Unterauer (2017), siehe Verzeichnis „Literatur zu Kapitel 4“.

Zur Diskussion des Statements Nr. 4 empfehle ich folgenden Artikel von Sam Newmann:

http://samnewman.io/patterns/architectural/bff/
(Last Access: 08.05.2022)

b.)
Modellieren sie eine *erste Version* einer neuen Software-Architektur auf Basis des Architekturstils Microservices. Modellieren sie eine Kontext-Sicht sowie eine Baustein-Sicht (vgl. Kapitel 3, Abschnitt 2). Falls schon anwendbar, können sie auch eine erste Verteilungssicht modellieren. Sie können sich bei der Baustein-Sicht gerne an die allgemeinen Erläuterungen zu Microservices aus Kapitel 2 (Abschnitt Microservices) sowie an die in Kapitel 3 (Abschnitt 4) orientieren. Modellieren sie auch eine *erste* Context Map (vgl. Kapitel 2 (Abschnitt Microservices))!

Bei der Modellierung der Sichten sollten sie sich zudem an die *abgeleiteten* Lösungen für *alle* Architecture Smells (Abschnitt IV.A) sowie für *alle* Anti Patterns (Abschnitt IV.B) aus dem Katalog von (Schirgi and Brenner, 2021) orientieren! Geben sie für jede Lösung eines Anti Patterns bzw. eines Architecture Smells eine erste Einschätzung ab, inwiefern diese für ihren *ersten* Architektur-Entwurf relevant sind und wie sie die Lösungen in die Software-Architektur integrieren konnten. Geben sie auch eine erste Einschätzung hinsichtlich Priorisierung, Aufwand sowie Implikationen für die spätere Implementierung einer jeden Lösung ab (vgl. ähnliche Herleitung wie bei der LZU, Übung Nr. 2).

Bei der Kontext-Sicht sollten auch die verschiedenen Akteure der Anwendung herauskommen. Sind die Akteure in der bestehenden IST-Architektur alle berücksichtigt worden? Tipp: Fokussieren sie sich bei der Baustein-Sicht sowie bei der Context Map auf die Komponente „Manufacturing Products“ – hier finden sie auch einige Hinweise dazu in der Architekturbeschreibung.