Zürcher Hochschule
für Angewandte Wissenschaften

School of
Engineering

Lernaufgabe

LE 11 – Architektur
Kap 4 – Lösungsstrategie formulieren

Aufgabe 16: Kapitel 4 der Architekturdokumentation (**2 Punkte)**

Lernziel

Sie können eine Lösungsstrategie für die Auktionsfallstudie gemäss Template Arc42 http://www.arc42.de/template/index.html erstellen.

Aufgabe 4 – Lösungsstrategie

Erstellen Sie das Kapitel 4 Lösungsstrategie beinhaltend eine Tabelle mit Architekturzielen und passenden Lösungsansätzen.

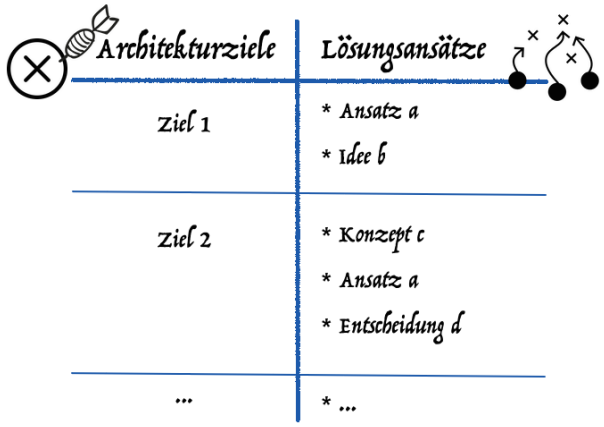
Überlegen Sie sich unter anderem, ob Sie die in der Lernaufgabe

Abgabe

Kapitel 4 der Architekturdokumentation in OLAT hochladen.

Hinweis

Eine besonders kompakte und wirkungsvolle Form zur Dokumentation und Kommunikation der Lösungsstrategie stellt die wichtigsten Anforderungen den Lösungsansätzen in einer Tabelle gegenüber, siehe folgende Abbildung:



Die linke Spalte enthält dabei die Qualitätsziele (oder auch Architekturziele, dieser Serie). Für ein ausdrucksstarkes Ergebnis zahlt es sich hier aus, wenn Sie den Zielen prägnante Namen gegeben haben (z.B. „Intuitive Erlernbarkeit“ statt nur „Benutzbarkeit“). In der rechten Spalte ordnen Sie den Zielen die wichtigsten Lösungsansätze Ihrer Architektur zu, die aus Ihrer Sicht der Erreichung der Ziele dienen. Als erstes sind hier die Architekturentscheidungen zu nennen.

Die folgende Tabelle listet mögliche Kategorien für Lösungsansätze für die rechte Spalte der Strategie auf und illustriert sie jeweils mit einem Beispiel. Die einzelnen Ansätze nennen Sie in Ihrer Tabelle nur schlagwortartig, beispielsweise „JPA/Hibernate als Persistenzframework“. Auf detaillierte Informationen (z.B. die ausführliche Darstellung einer Architekturentscheidung inklusive betrachteter Alternativen, das ausgearbeitete Konzept etc.) verweisen Sie lediglich.

Kategorie	Beispiel (und dazu passendes Qualitätsziel)
Entscheidungen	Verwendung eines Application Server Clusters (hohe Ausfallsicherheit)
(Architektur-)stile	Micro Services (schnelle Adaption neuer technologischer Trends)
(Architektur-)muster	Schichtenarchitektur (leichte Austauschbarkeit des Clients, oder einfache Portierung der Lösung)
(Architektur-)prinzipien	Bevorzuge Standards vor proprietären Lösungen (niedrige Wartungsaufwände)
Konzepte	Caching-Konzept (Effizienz, gute Antwortzeiten)
Vorgehen	User centered design (intuitive Benutzbarkeit)

In Einzelfällen können Sie aber auch Randbedingungen als „Argumente“ für Ihre Architektur anführen. Wenn beispielsweise Technologien vorgegeben sind, die gleichzeitig gut zu den Zielen passen, können Sie diese in der rechten Seite ebenfalls nennen. Ein einzelner Lösungsansatz kann mitunter mehreren Zielen dienlich sein. Sie listen ihn dann einfach mehrmals in der rechten Spalte auf.

Die folgende Tabelle stellt den Qualitätszielen von Gradle passende Lösungsansätze (rechts) gegenüber.

Qualitätsziel	**passende Lösungsansätze in Gradle**
Erweiterbarkeit (Gradle lässt sich leicht um neue Funktionalität erweitern. Es kann auf lange Sicht dem technologischen Fortschritt bei Tools und Entwicklungsmethodik folgen)	• Build-Files sind Groovy-Skripte, kleine Erweiterungen auch ohne aufwändige Programmierung möglich • Ausgereiftes Plugin-Konzept, Custom Plugins können in Groovy, Java, Scala … entwickelt werden • Exzellent dokumentierte Gradle API und erweiterbare DSL
Effizienz (Teams und einzelne Entwickler erhalten durch kurze Buildzeiten schnelle Rückmeldungen; Gradle steigert so ihre Produktivität)	• Unterstützung inkrementeller Builds • Dependency-Cache zur Reduktion der Download-Zeiten • Betrieb im Hintergrund für kürzere Start- und Ausführungszeiten möglich (Gradle Daemon)
Interoperabilität (Gradle arbeitet mit bestehenden Werkzeugen wie Ant und Maven und deren Öko-Systemen nahtlos zusammen)	• Direkte Verwendung von Ant-Tasks und Ant-Projekten in Gradle möglich • Konverter für Maven pom.xml nach Gradle Build-Skript verfügbar • Unterstützung von Maven-Repositories (Download von Abhängigkeiten, Veröffentlichen von Artefakten) • Gradle API erlaubt Einbetten von Gradle, zum Beispiel in IDEs