

ID	Funktionale Anforderung (aus Sicht eines Component Assemblers)	Priorität (MusCoW)	Aufwand (h, m, g)	Notwendige Pattern	Notwendige (Java-)Techniken	Anmerkungen
1	Management von Abhängigkeiten von Komponenten (Analyse, Auflösen)	C-W		Dependency Injection		
2	Deployment von Komponenten (Einsetzen von Komponenten in eine LZU)	M	sh		Classloading; generell OO; Reflection API; String Handling	
3	Start / Ausführung von Komponentne	M	m-h		Java Annotations	
4	Konfiguration von Komponenten	M-C				
5	Status-Abfrage über den Lebenszyklus und Meta-Daten	M		State Pattern?		Versionangaben; Beschreibung, Libraries, Name, Author, Interfaces
6	Komponenten entfernen	M				
7	Komponenten stoppen	M				
8	Error-Handling bei Abhängigkeiten (z.B. bei fehlenden Abhängigkeiten)	C - W				
9	Bereitstellung von Diensten (Services) wie z.B. Caching oder Logging	C				
10	Starten der LZU / Konfiguration	M	m-h			
11	Stoppen der LZU	M				
12	Monitoring von Komponenten (ggf. der LZU)	C-W				
13	Persistierung des Zustands (Sicherung, Recovery)	C				
14	Authentication und Authorization des Component Assemblers	C-W				
15	Nebenläufigkeit von Komponenten (i.e.S. das parallele Starten und Ausführen von Komponenten)	M			Synchronisierung; Java Threads	ist zum Teil "deprecated" (Stop eines Threads)
16	Komposition von Komponenten					
17	Management-Schnittstelle (CLI; Web-Interface)	M		Command		

**Legende Prio:**

M = Übung Nr. 2

C = Weitere Übung ab Nr. 3

W = Nice to Have, ggf. für Semesterprojekt relevant