

	Gewicht	Punkte	Note	Bemerkungen
<b>Anforderungen</b>	20%	2.3	5.3	
Ermittlung	1 33%	2	0.7	Methodik nicht gut erkennbar, Sitzungen etwas salopp, keine Protokolle (macht es schwierig die Abläufe nachzuvollziehen)
Dokumentation	1 33%	2.5	0.8	Die Artefakte sind im README etwas versteckt oder müssen selbst gesucht werden (e.g. SRS, da gibt es dann aber zwei Versionen...). Die Abschnitte des SRS sind etwas seltsam ausgefüllt (3.1. Externe Schnittstellen Anforderungen, und dann 3.2. Funktionale Schnittstellen, wa ist das?). Etwas inkonsistent (es gibt z.B. zwei Benutzergruppen, aber im Use Case Diagramm ist nur ein Akteur sichtbar)
Umsetzung	1 33%	2.5	0.8	Priorisierung und Iterationen nicht gut erkennbar, Tendenz zu Wasserfall mit Big Bang, erst am Schluss eine Feedback Runde gemacht
	3 100%			
<b>Frontend</b>	20%	3.0	6.0	
Funktionen, Basis	1 33%	3	1.0	Alles da
Funktionen, Zusatz	1 33%	3	1.0	Mehrfachselektion, Verwendung der Tags
Benutzbarkeit	1 33%	3	1.0	Einfach zu erlernen, effizient zu benutzen, sieht ansprechend aus
	3 100%			
<b>Architektur</b>	20%	2.5	5.5	
Micro-Services	1 50%	2	1.0	fest vorgegebene URL, CORS Konfiguration in nginx.conf
Interfaces (z.B. REST, Datenbanken)	1 50%	3	1.5	DB Schema mit ddl-auto
	2 100%			
<b>Plattform</b>	20%	3.0	6.0	
Administration / Konfiguration des Servers	1 33%	3	1.0	Ok, Verzeichnisse aufgeräumt
https-Zertifikat	1 33%	3	1.0	Ok
PACS / DICOM	1 33%	3	1.0	Ausgeklügelter Importer, Daten werden geholt, alles ok
	3 100%			
<b>Konstruktion</b>	20%	2.5	5.5	
Git / Issue Tracking	1 25%	2	0.5	unterschiedliche und eigenartige Autorennamen (marc stawiski, marc.stawiski, Marc Stawiski), keine Issue-Refs, viele temporäre Branches mit unklarer Bezeichnung
Automatischer Build-Prozess	1 25%	3	0.8	erfüllt
Coding Style, Error Handling, Logging	1 25%	2	0.5	wenig logging
Unit Tests / Reporting	1 25%	3	0.8	keine frontend tests
	4 100%			
				Punkteskala: 3 - fully 2 - largely 1 - partially 0 - not
Gesamtpunkte (gewichtet)		2.7		
<b>Gesamtnote</b>	100%	5.7	5.67	