

## Checkliste MS 4: Review Design

Seite 1

Review Design (Gewicht ¾)		Note:						
Fazit:								
		++	+	o	-	--		
Architektur/Design inhaltlich								
	Physische Architektur (Verteilung auf Rechner, Prozesse): Schwierigkeitsgrad? Gut gelöst?							
	Logische Architektur (Gliederung in Packages, Schichtenarchitektur): Schwierigkeitsgrad? Gut gelöst (Kohäsion)?							
	Schnittstellen zwischen Packages: Gut gelöst (geringe Kopplung)?							
	Persistenz: Gut gelöst?							
	Design der Packages: Gute Klassenstruktur (eventuell Designpattern, wo sinnvoll)?							
	Externes Design: Gut gelöst?							
	Besonderheiten des Designs: gut gelöst?							
	Eingesetzte Technologien: Schwierigkeitsgrad, gut gemeistert?							
Architektur/Designdokumentation								
	Vollständig gemäss Musterinhaltsverzeichnis auf folgender Seite?							
	Korrekt, gut verständlich?							
	Das Wesentliche beschrieben?							
	Designentscheidungen begründet?							
	Form, Sprache, Stil							
Stand der Arbeiten (Gewicht ¼)		Note:						
Fazit:								
		++	+	o	-	--		
	Zeiterfassung nachgeführt und ausgewertet auf einzelne Teammitglieder							
	Stand der Arbeiten gemäss Projektplan /							
	Stand der Arbeiten gemäss Semesterwoche							
	Beiträge der einzelnen Teammitglieder ausgewogen							
	Arbeitsergebnisse gemäss Feedback letzter Review überarbeitet							
	Alle Dokumente an dieser Review aktuell und einsehbar							
Gesamtbeurteilung am Meilenstein		Note:						
Fazit:								

Legende: ++ sehr gut / + gut / o genügend / - schlecht / -- sehr schlecht oder fehlt

**Checkliste MS 4: Review Design**

Seite 2

**Dokumentation des Entwurfs: Musterinhaltsverzeichnis aus SE1****1. Einführung**

1.1 Zweck, 1.2 Gültigkeitsbereich, 1.3 Definitionen und Abkürzungen, 1.4 Referenzen

**2. Software-Systemarchitektur****2.1 Architekturübersicht**

- Einführende Beschreibung der Architektur (oft mit informalem Architekturdiagramm, das meist physische Sicht oder Mischung von physischer, logischer Sicht darstellt)

**2.2 Systemstruktur**

- Physische Sicht:
  - Deployment-Diagramm zeigt Verteilung auf einzelne Tiers
  - Kurze Beschreibung und Begründung für gewählte physische Architektur
- Logische Sicht:
  - Diagramm mit Packages und Dependencies zwischen Packages
  - Kurzbeschreibung der Packages
  - Eventuell Kurzbeschreibung wesentlicher Klassen (Exportklassen, Facades)
  - Einzelne Packages sind z.T. Technologiekomponenten wie GUI-Library
    - Kurze Beschreibung der gewählten Technologiekomponenten mit Begründung für ihre Wahl (ev. Verweis auf einen Anhang, der Varianten diskutiert)

**2.3 Schnittstellen der Packages**

- Exportklassen der Packages mit allen public-Methoden
- Architektur der Interaktion (z.B. zwischen Domain GUI: MVC-Architektur),  
Ev. Illustration mit Interaktionsdiagrammen

**2.4 Architekturkonzepte**

- Generell angewandte Lösungsmuster, die nicht einem Package zugeordnet werden, wie Errorhandling, Logging- und Debugging-Mechanismen, Speicherverwaltung u.a.m. mit Begründungen für gewählte Lösung
- Multitasking und Multithreading, d.h. Process View: aktive Objekte (wenn nötig)

**3. Beschreibung der Packages****3.1 Package *Name*****3.1.1 Klassenstruktur**

- Klassendiagramm mit Design-Information wie Zugriffsebenen, Navigationspfaden, Containment (By Value, By Reference)

**3.1.2 Architekturkonzepte für Package (falls vorhanden)**

- Umsetzung der generellen Architekturkonzepte
- Spezifische Architekturkonzepte für Package

**3.1.3 Klassenspezifikationen**

- nur wo sinnvoll, insbesondere Schnittstellenklassen
  - mit wesentlichen Attributen und Methoden (meist als Javadoc)
  - Eventuell Datentypen für Attribute, Parameter und Rückgabewerte mit Datentypen für Methoden, eventuell Zustandsdiagramm für Klasse (falls sinnvoll)

**3.1.4 Interaktionen innerhalb Package (falls sinnvoll)****3.2 Package *A*****3.3 Package *B***

...

**4. Real Use Cases**

- Interaktionsdiagramme für Use Cases bzw. Systemoperationen
- Wenn nur Interaktionen, in einem Package gezeigt (z.B. Domain), dann auch als 3.1.4

**5. ...Ev. Weitere Beschreibungselemente****Anhang:**

- z.B. Diskussion Design-Alternativen

**Detailbemerkungen:**

...

- ...