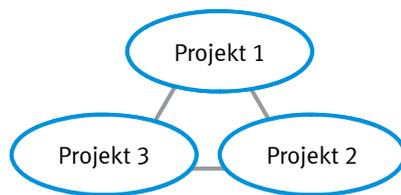


# Vorlesung Software-Engineering I

... im 3. und 4. Semester

## Projektgruppen – Organisation 3. Semester



## Inhaltsübersicht Software-Engineering I - **Noten**

Die Vorlesung dauert **2 Semester** (3. und 4. Semester)

Es gibt nur **eine Note** über beide Semester

**9 ECTS Creditpoints**

### 3. Semester:

Test (30 Min.)	1/3	Vorgehensmodelle
Projektarbeit	2/3	Anforderungsanalyse + Architekturentwurf

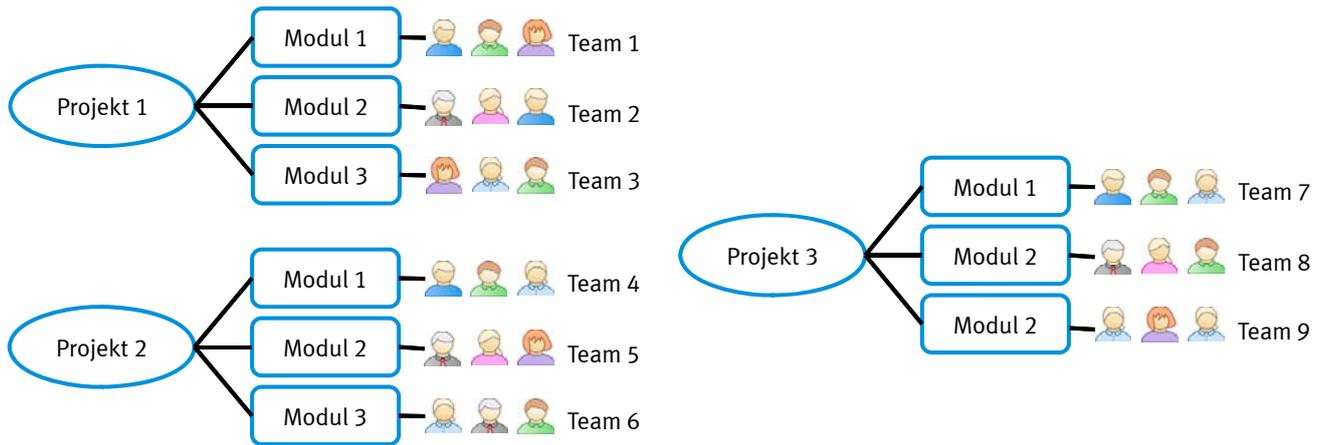
### 4. Semester:

Test (30 Min.)	1/3	Basistechniken (UML)
Projektarbeit	2/3	Architekturentwurf, Umsetzung +Testmanagement

# Inhaltsübersicht Software-Engineering I - Gruppenarbeit

In jedem Semester werden **mehrere Projekte** bearbeitet und bewertet. Die Bearbeitung und Bewertung bricht sich dabei auf einzelne **Module** und **Teams** herunter. Es wird zwischen der **Gruppenleistung** und der **Teamleistung** bei der Bewertung unterschieden.

Die Bewertung erfolgt anhand den **Phasendokumente** wie Lasten-/Pflichtenheft, SW-Architektur unter Anwendung der entsprechenden **Methoden** und **Darstellungen**.

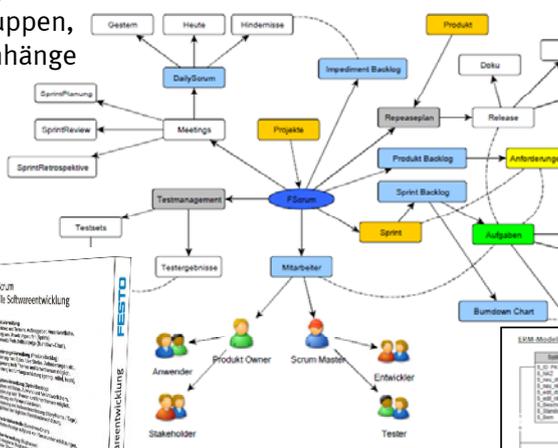


Exemplarische Ansicht!

# Artefakte: Idee -> Übersicht -> Anforderungen -> Software-Architektur

## Konzept-Map:

-> Themengruppen, Zusammenhänge



## Anforderungsliste:

-> Anwendungsfälle, Anforderungen, Ideen

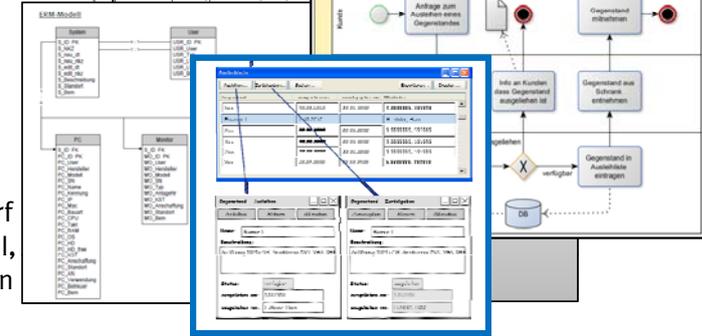
Thema	Typ	Wort-Stich oder Anforderung	Vorgeschlagene Umsetzung / Anmerkungen	Wart.	Archiv.	Spezif.	Ablauf	Verfügb.
Forum	Epic	Das Forum Tool soll folgende Komponenten besitzen: Profilverwaltung, Anzeigenverwaltung (Produkt Backlog), Aufgabenverwaltung (Sprint Backlog), Aufgabenverwaltung (Burndown Chart), Bugverwaltung, Teamverwaltung.						
Forum	Epic	Die Abhängig (SM) sollten aus bisher verwendeten Modulen auf Basis von Forum System durch eine Erweiterung entstehen.						
Forum	Epic	Das Forum Tool soll dem agilen Scrum-Prozess angepasst sein und unterstützen.						
Forum	UserStory	Der Benutzer möchte mithilfe der Tool die Anzeigen/Produkt Backlog, Sprint Backlog, Burndown Chart und Team Methoden verwalten.						
Forum	Epic	Das Tool sollte Funktionen wie Bugtracking oder Versionsverwaltung beinhalten (Epic).						
Benutzerberichte	Anforderung	Die Benutzerberichte sollte einfach zu bedienen sein und möglichst leichtklickbar sein.						
Rechte	UserStory	Der Benutzer möchte sich beim Start des Tool mit einem Kurztesten und Gruppen/Projekt anmelden können.						
Forum	Anforderung	Das Tool sollte einen virtuellen Team Chat besitzen.						
Forum	Anforderung	Das Tool soll mit der Entwicklung kompatibel sein.						
Forum	Anforderung	Die Entwicklung soll Plattformunabhängig sein.						
Forum	Anforderung	Die Benutzer soll geschäftig an den						



**Produktkarton:**  
-> Zielgruppe, Leistungsmerkmale

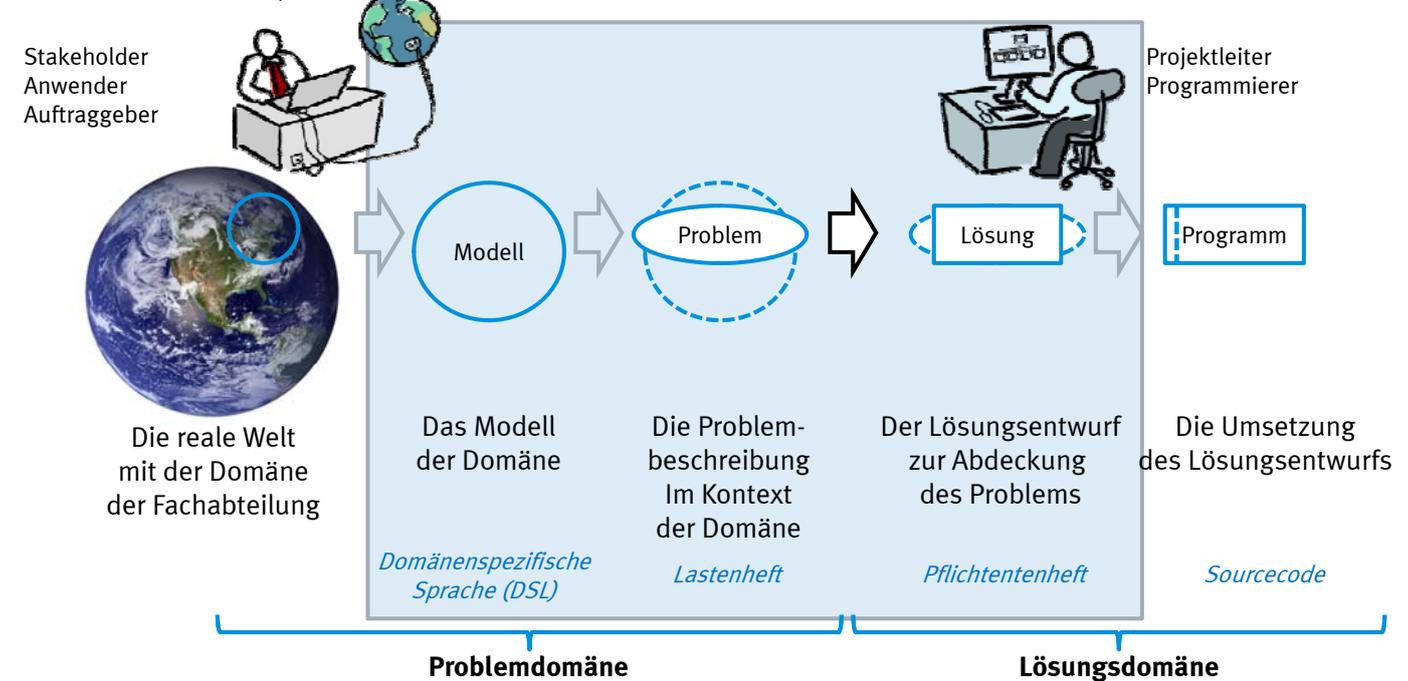
## SW-Architektur:

-> Umsetzungsentwurf, Module, Daten, GUI, Abläufe, Funktionen

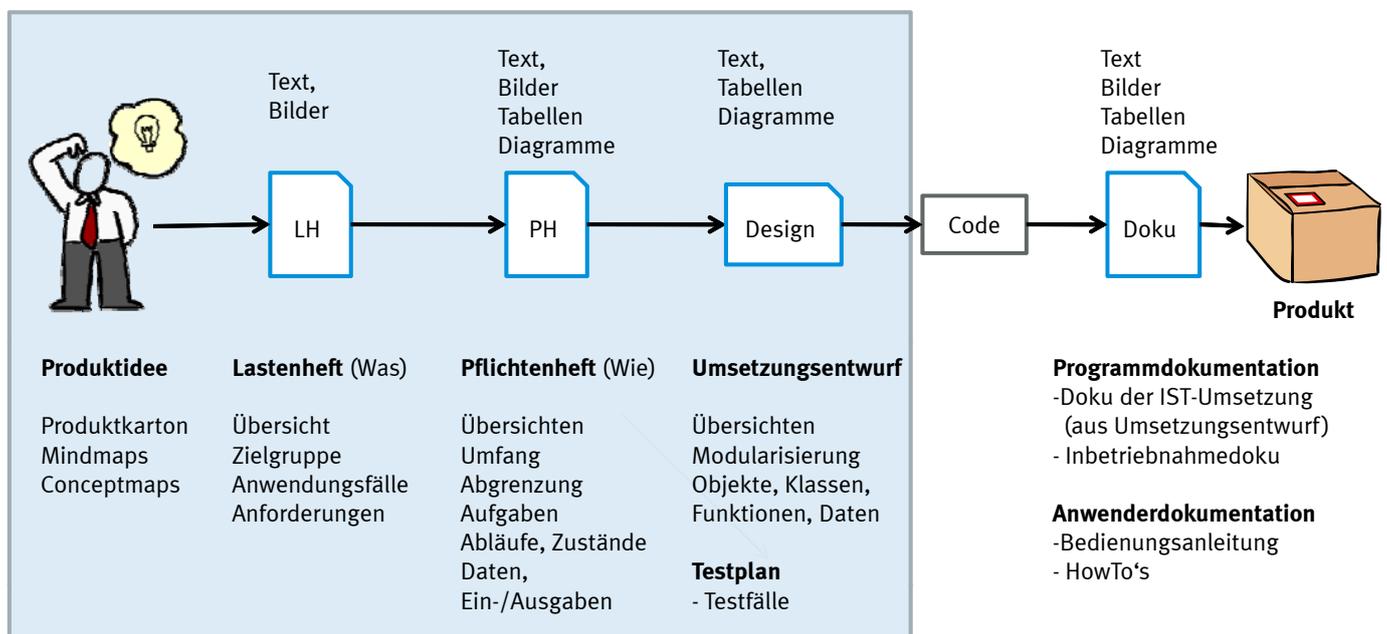


## Vom Problem zur Lösung

oder: Domänen und ihre Sprachen

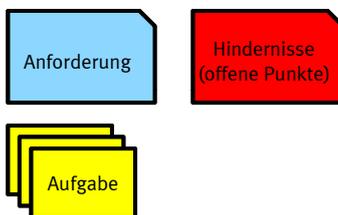


## Artefakte und ihre Inhalte

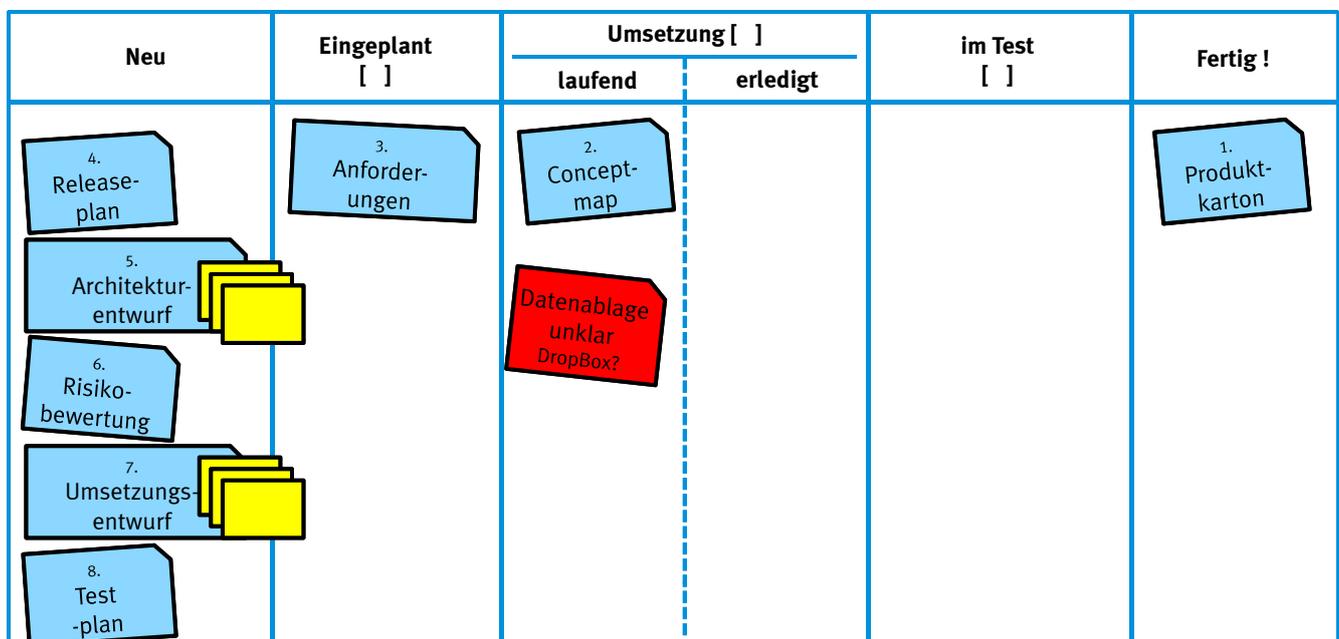


## Aufgaben/**Inhalt**/Ergebnis:

<b>1. Produktkarton</b> erstellen	- Zielgruppe - Leistungsmerkmale - Rahmenbedingungen - Logo	-> <b>Produktkarton</b>
<b>2. Concept Map</b> erstellen	- Themen, Module - Rollen (Personen), Daten - Verknüpfungen	-> <b>Concept Map</b>
<b>3. Anforderungen</b> aufnehmen	- Anwendungsfälle - Anforderungen (funk./n.funk.) - Rahmenbedingungen	-> <b>Produkt-Backlog</b> -> <b>Lastenheft</b>
<b>4. Releaseplan</b> erstellen	- Versionen, Umfang (prod./n.prod) - Termine	
<b>5. Architektur-Entwurf</b> erstellen	- Workflows (Rollen, Daten-Stat, Funktionen) - Module, Daten - Benutzeroberfläche	-> <b>Architektur-Dokumentation</b> -> <b>Architektur-Präsentation</b>
<b>6. Risiko-Bewertung</b> durchführen	- Review („Durchstich“ prüfen)-> Prototyping nötig? - evtl. Refactoring des Entwurfs	-> „ <b>offene Punkte</b> “-Liste
<b>7. Umsetzungs-Entwurf</b> erstellen	- Abgrenzung (Scoping) - Module, Funktionen, Daten	-> Umsetzungs-Entwurf
<b>8. Testplan</b> erstellen	- Testplan	-> Testplan



## Projektstatus: Das Kanban-Board



## Datenablage



[hoyer@lehre.DHBW-Stuttgart.de](mailto:hoyer@lehre.DHBW-Stuttgart.de)

<http://wwwlehre.dhbw-stuttgart.de/~hoyer/>



DropBox.com  
„Shared Folder“



drive.google.com  
„Shared Folder“

[festo.hoy@googlemail.com](mailto:festo.hoy@googlemail.com)

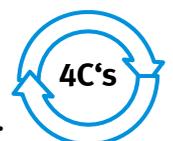
## Rahmenbedingungen

1. **Namensgebung der Dokumente** beachten:  
[<Projekt>\\_<Modul>\\_<Dokumenttyp>\\_<Datum>\\_<Ersteller>.ext](#)  
 z.B. Prj1\_Konfig\_ConceptMap\_2015-09-19\_FMH.pdf  
 (immer auch eine PDF-Version erstellen)

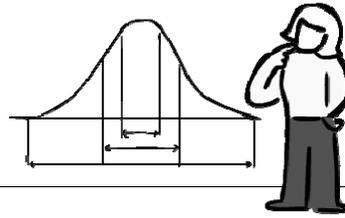


2. **erledigte Dokumente** einstellen.  
 -> Nach dem Einstellen bekommen man ein Feedback mit evtl. Verbesserungsvorschlägen

3. es kann jederzeit ein **Update der Dokumente** geschickt werden!  
 Dies ist explizit erwünscht. Vor allem als Reaktion auf das Feedback.  
 Evtl. ein Changelog mitschicken in dem beschrieben wird was warum geändert wurde.

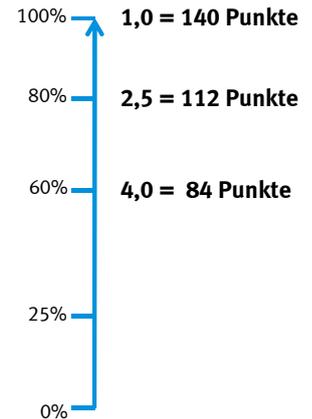


4. Es können auch **nicht spezifizierte Dokumente** geschickt werden die den Projektverlauf dokumentieren. z.B. Besprechungsprotokolle, Statusberichte.



## Vereinfachte Punktbewertung

SWE1 Bewertung der Projektarbeit		max. 10 Punkte pro Aufgabe möglich				
3.Semester 2011		Projekt #, Team #				
Aufgaben	Artefakt	Gewichtung	Strukturiertheit	Vollständigkeit	Verständlichkeit	Summe Punkte
01-Produktidee	Produktkarton	1	2	2	3	7
02-ConceptMap	ConceptMap	1	2	2	3	7
03-Anforderungen	Produkt-Backlog	1	2	2	3	7
04-Modularisierung	Protokoll	1	2	2	3	7
05-Lastenheft	Lastenheft	1	2	3	3	8
06-Datenmodell	ER-Modell	1	2	3	3	8
	DataDictionary	1	2	3	3	8
07-Abläufe	BPMN	1	2	3	3	8
	Aufruf-Hierarchie	1	2	3	3	8
08-Benutzeroberfläche	Mock-up	1	2	3	3	8
	Offene-Punkte-Liste	1	2	3	3	8
09-Review	Offene-Punkte-Liste	1	2	3	3	8
10-Dokumentation	Dokumentation (Print)	1	2	3	3	8
	Präsentation	1	3	3	4	10
Sonstiges	Eigene	1	3	3	4	10
	100%	Note 1,0 bei:	140	Summe:	112	
	60%	Note 4,0 bei:	84	Note:	2,5	2,5 bei 80%



## Bewertungskriterien

### Strukturiertheit:

Sind die Vorgaben berücksichtigt worden?  
(Layout, Kapitel, Aufgabenstellung, etc.)

### Vollständigkeit:

Ist die Darstellung vollständig?  
(das Wichtigste dargestellt, offene Punkte/Vereinfachungen dokumentiert)

### Verständlichkeit:

Ist die Darstellung verständlich?  
(Zielgruppenorientiert im Format einer Präsentation)

### Wichtig:

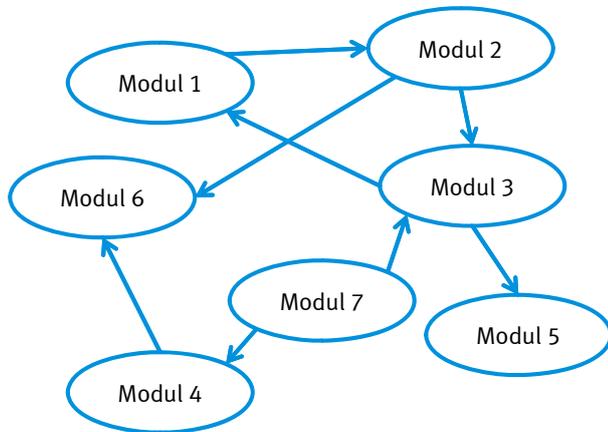
Sie sollen in der Vorlesung etwas über Software-Engineering und Projektarbeit lernen;  
setzen Sie sich mit dem Thema auseinander... Fehler sind erlaubt!

**-> Fehler erkennen und Verbesserungsvorschläge machen!**

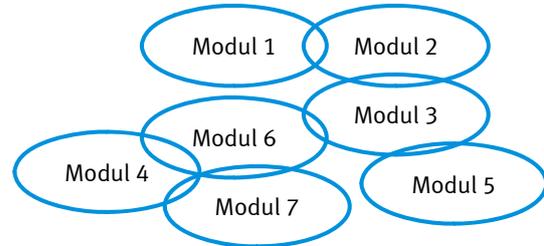




## Teamarbeit: Schnittstellen oder Schnittmengen?



- + definierte Schnittstellen
- + klare Aufgabentrennung
- + einfache Bewertung
- Probleme bei Änderungen  
an den Anforderungen/Schnittstellen
- Aufwand für Absprachen



- + definierte Abhängigkeiten
- + klare Aufgabenstellung
- + gemeinsame Erarbeitung der Schnittstelle
- + offen für Änderungen
- schwierige, aufwändige Bewertung
- der Code gehört nicht mehr mir

## Fragen:

