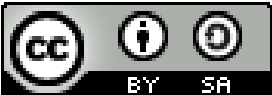


Agiles Software-Engineering

Prof. Dr. Gerd Beneken

Woche 02, Kapitel 6


Kontext verstehen - Stakeholder



Verstehen der beteiligten Personen

■ **Kunden kennen lernen:**

- Wer sind die Beteiligten (**Stakeholder**)?
- Wie ist der Kunde organisiert?
- **Wer gibt das Geld aus? (der entscheidet)**
- Wer hat welchen Einfluss, wie sind die sozialen Beziehungen?



Stakeholders are people who have an **interest** in the product

■ **Ansprechpartner finden**

■ **Nutzer kennen lernen:**

- Welche Personen werden das System nutzen?
- Was tun diese Personen (Job)?
- Wie sind sie qualifiziert?
- Was ist für sie wichtig?
- Aufschreiben: Personenbeschreibung
- **-> Personas**

Projektbetroffene Stakeholder

■ engl: *Stakeholder*:

„Stakeholders are people who have an interest in the product“

Offensichtlich:

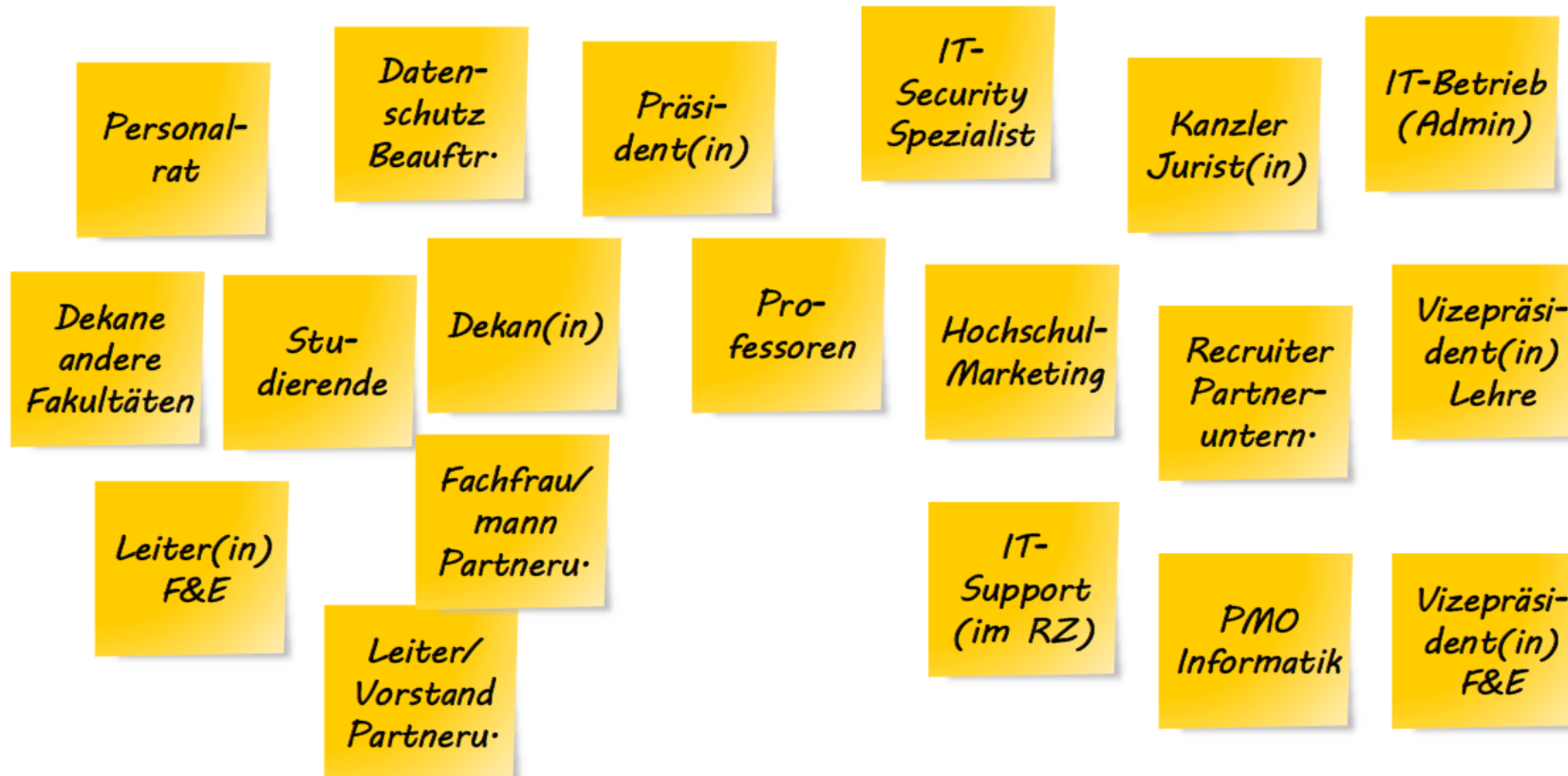
- Auftraggeber
- Benutzer (End-User)
 - regelmäßige, gelegentlich
 - direkt, indirekt
- Fachabteilung

Nicht vergessen:

- Datenschutzbeauftragte:r
- DV-Betrieb (z.B. Operating, Ops)
- Methodenabteilung (DV-Koordinatoren)
- Technischer Support
- Hotline / Helpdesk
- Betriebsrat
- Marketing (bei Produkten)
- Entwickler des Kunden
- Nachbarsysteme

Stakeholder Brainstormen

STAKEHOLDER



Beispiel für Ansprechpartner-Liste (Im Projekthandbuch / Projektwiki)

<i>Ansprechpartner</i>	<i>Rolle</i>	<i>Kontaktdaten</i>	<i>Kommentar</i>
<i>Allesandro Dante</i>	<i>Geschäftsführung</i>	<i>089/5557445</i>	<i>Auftraggeber für das Softwaresystem</i>
<i>Horst Pestalozzi (Stud.)</i>	<i>System- administration</i>	<i>089/5558965</i>	<i>Spricht nur englisch</i>
<i>Silvio Boccaccio</i>	<i>Buchhalter</i>	<i>08031/555827</i>	
...

Wichtig: Liste muss für jedes Teammitglied
Jederzeit greifbar sein! Im Projektwiki, Projekthandbuch

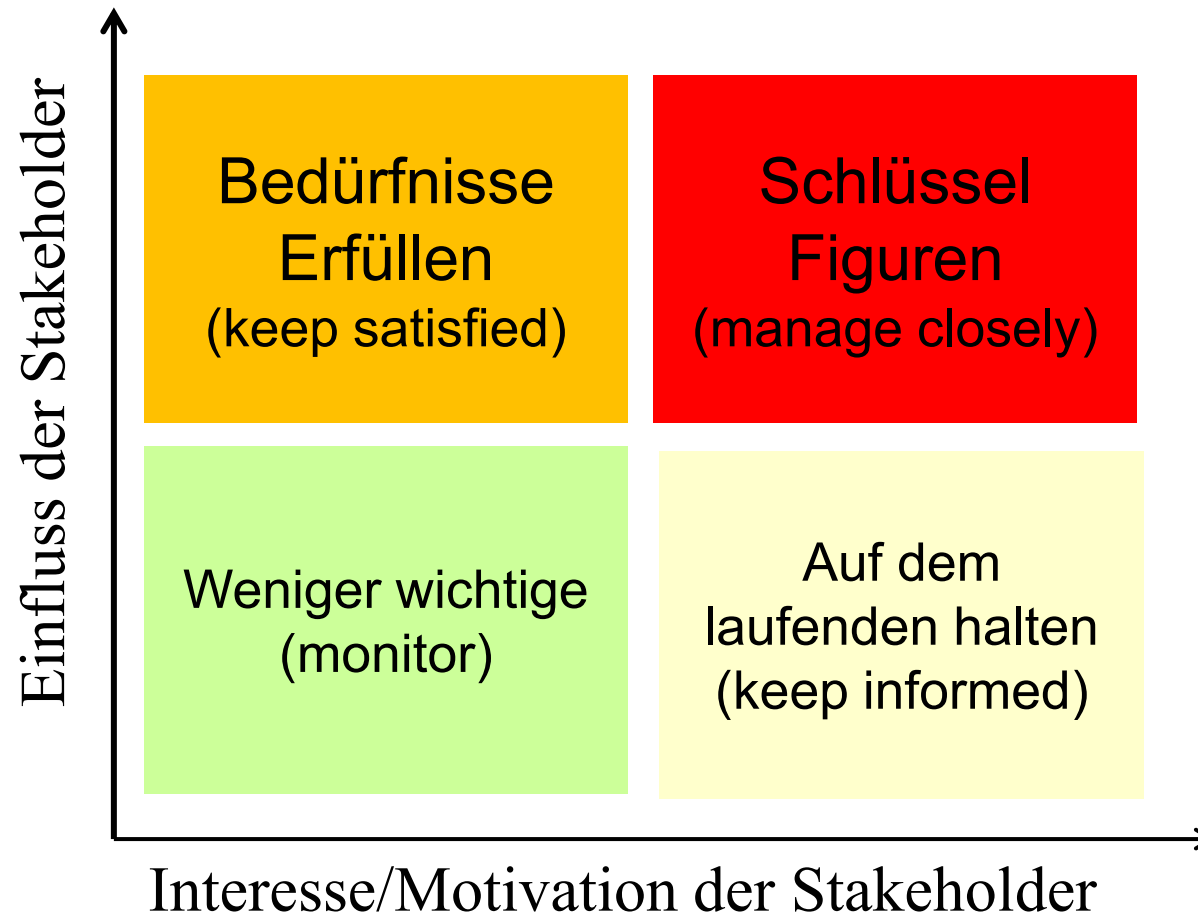
Beispiel für Stakeholder / *Interessen* Liste (ggf. im Projekthandbuch oder Projektwiki)

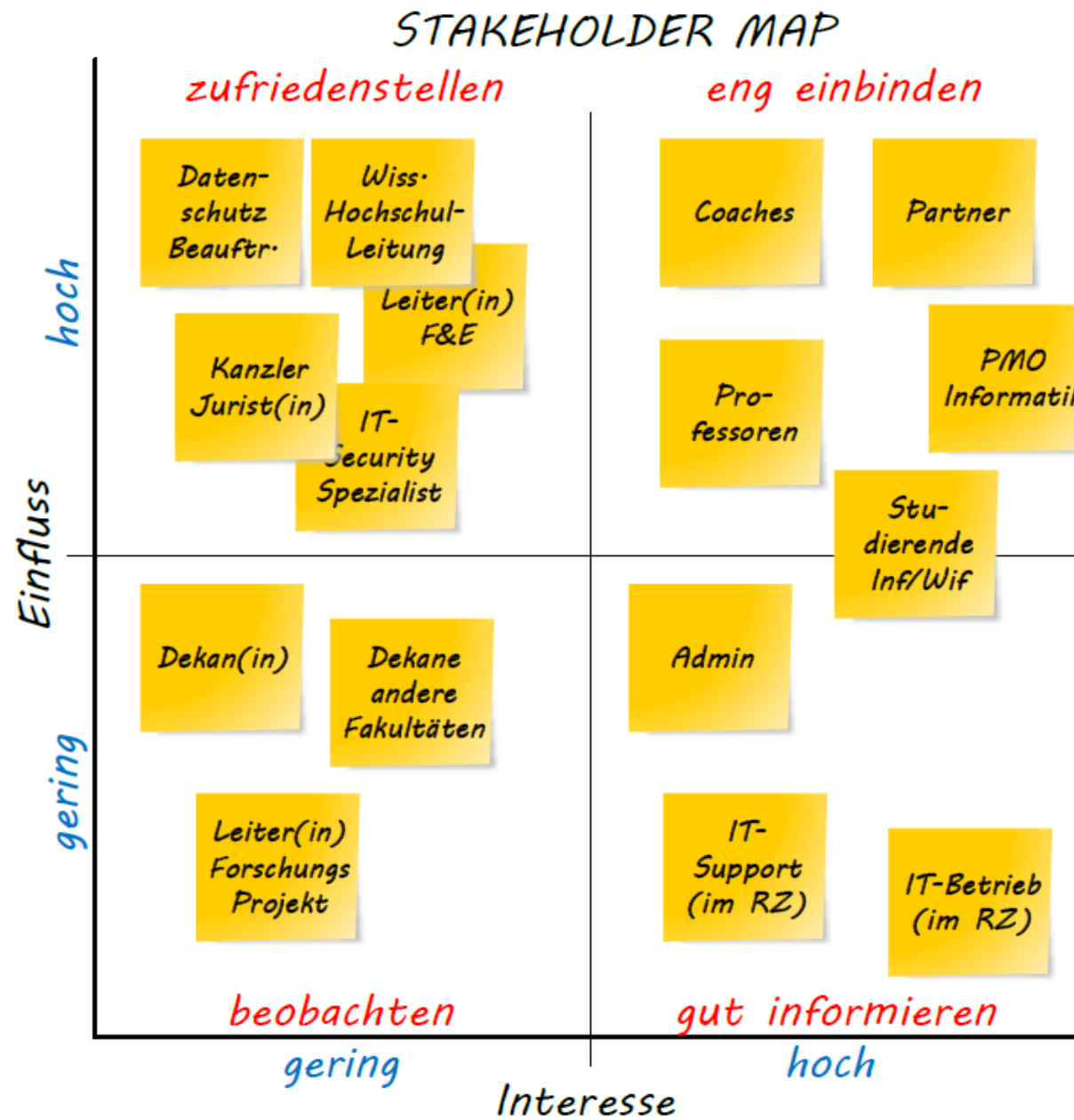
Gruppe	Interesse / Ziele	Vertreter
<i>Geschäftsführung</i>	<ul style="list-style-type: none"> • <i>Neuer Vertriebskanal</i> • <i>Mehr Umsatz</i> 	Hr. Dante
<i>System-administration</i>	<ul style="list-style-type: none"> • <i>Einfachere Bedienung, Arbeitszeiterparnis</i> • <i>Höhere Verfügbarkeit durch automatisierte Administrationstätigkeiten</i> • <i>Cloud wäre optimal</i> 	Hr. Pestalozzi
<i>Buchhaltung</i>	<ul style="list-style-type: none"> • <i>Weniger manuelle Eingriffe</i> • <i>Automatische Erzeugung der Bilanz</i> 	Hr. Boccacio
<i>Koch</i>		N.N.
<i>Fahrer</i>	...	N.N.

-> Wann genau ist das Projekt ein Erfolg?

Fehlende Ansprechpartner Identifizieren

Denkwerkzeug: Stakeholder Map

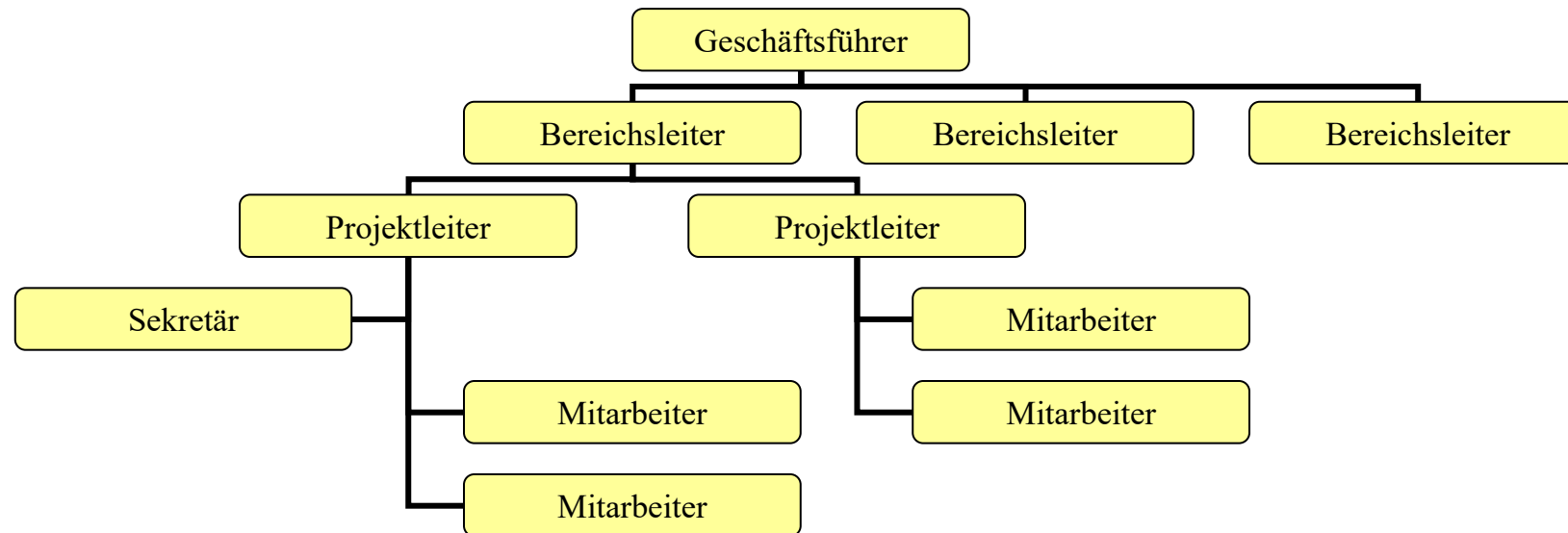




Kommunikationsplan ableiten

Person, Rolle oder Gruppe von Stakeholdern (Wer?)	Informationen (Was?)	Wie?	Turnus (Wann?)	Wer kommuniziert?	
	<i>Statusbericht, neue Features, eskalierte Probleme, nur Termine, ...</i>	<i>Persönliche E-Mail-Verteiler, Nachricht, Präsentation, persönliches Gespräch</i>	<i>E-Mail, Chat-Meeting, persönlich</i>	<i>Täglich, Wöchentlich, Monatlich, Quartalsweise, bei jedem Meilenstein, nach jedem Sprint, ...</i>	<i>Product Owner, das Team, Beauftragte für bestimmte Themen, ein Teammitglied</i>
Projektleitung	Statusbericht	Persönliche E-Mail	Wöchentlich	Product Owner	
Projektleitung	Projektfortschritt	Sprint Review Meeting	nach jedem Sprint	Team	
Rechenzentrum	Technische Änderungen	Meeting des DV-Ausschusses	Quartalsweise	Product Owner	

Denkwerkzeug: Organigramm



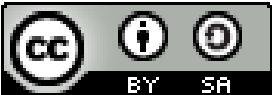
- macht Hierarchie des Unternehmens deutlich
- zeigt „Befehlswege“ / Verantwortlichkeiten / Eskalationswege
- zeigt ggf. Ressorts / Fachbereiche
- nennt Ansprechpartner

Agiles Software-Engineering

Prof. Dr. Gerd Beneken

Woche 02, Kapitel 7

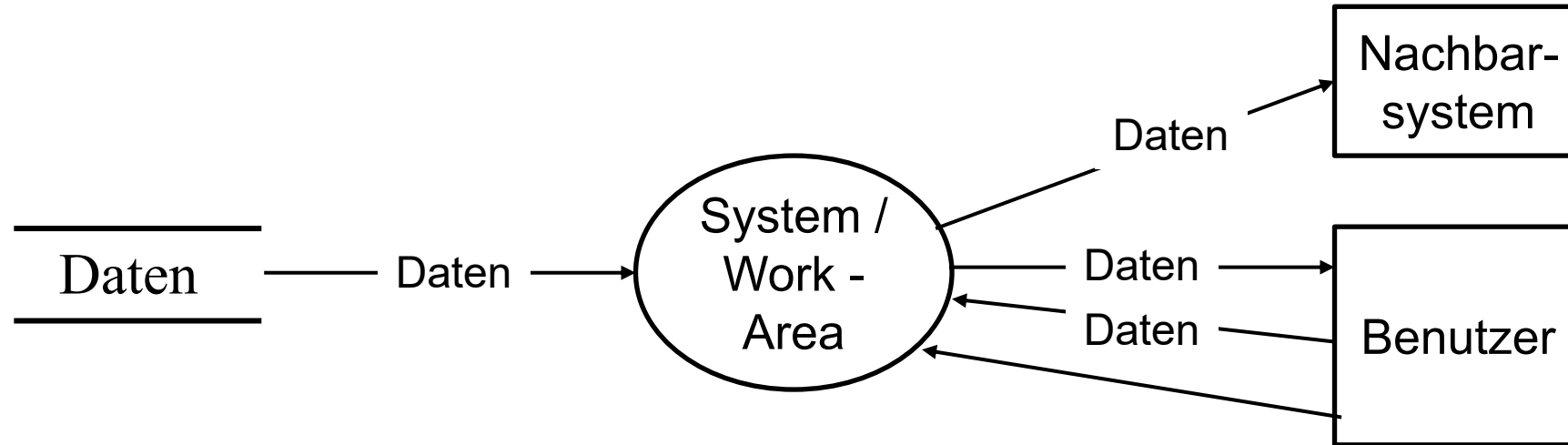
Kontext verstehen - Umgebung



Verstehen der Systemumgebung

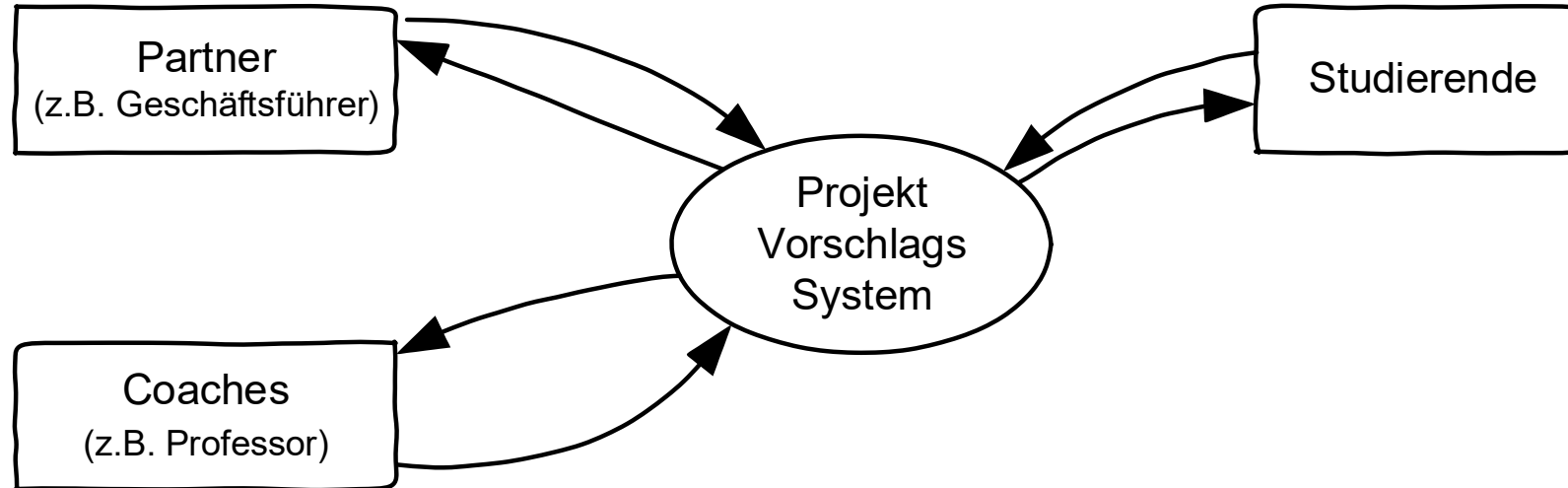
- Welche Teile der Geschäftsprozesse sollen im neuen System ablaufen?
(= **Scoping**)
- Wer sind die potentiellen Benutzer des Systems?
 - Welche Daten werden mit diesen Benutzern ausgetauscht?
- Welche **Nachbarsysteme** gibt es, und was machen diese?
 - Wer sind die Verantwortlichen für diese Systeme (mit denen müssen Sie reden)
 - Welche Releaseplanung haben die Nachbarsysteme (wenn Sie Änderungen brauchen)
 - Welche Daten tauschen Sie mit diesen aus?

Denkwerkzeug: Datenflussdiagramm - Notation



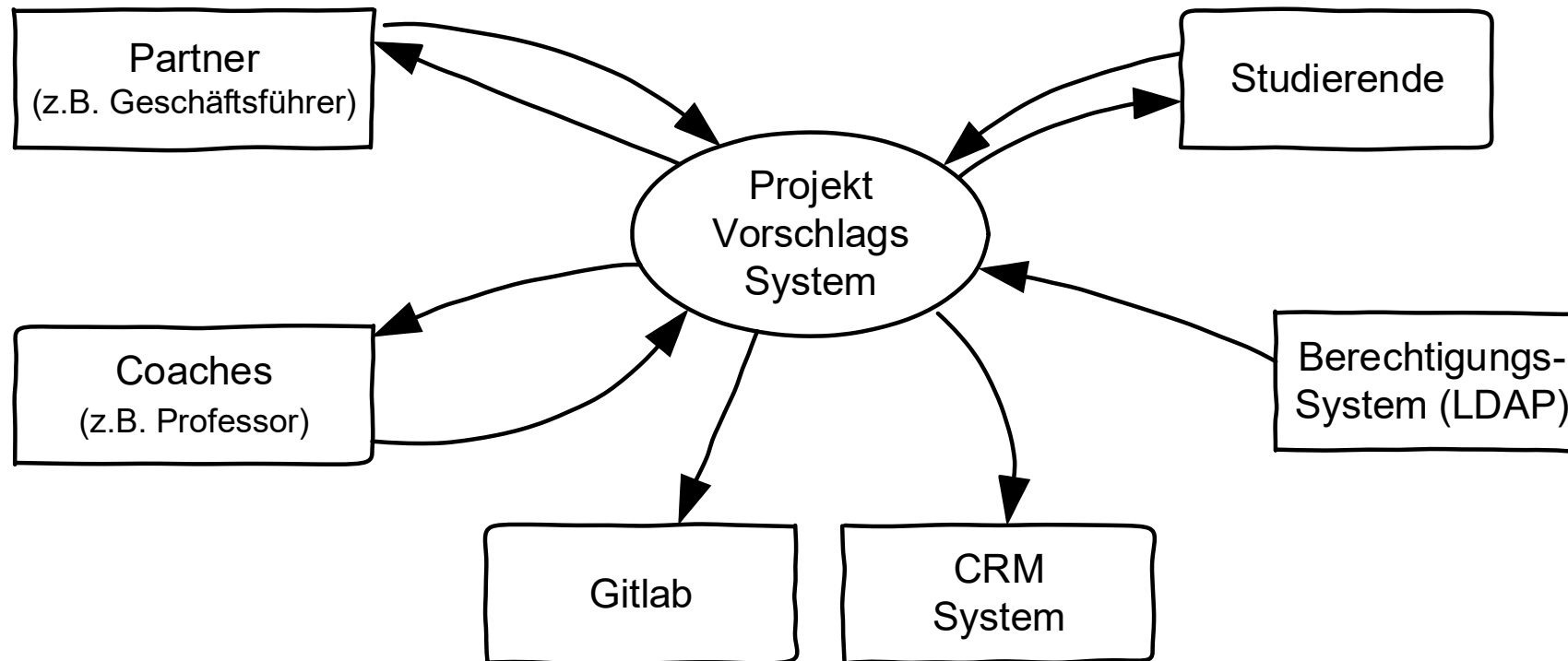
- Modelliert Datenfluss über beschriftete Pfeile
- Input/Output: Nachbarsysteme, Benutzer, Datenspeicher
- Ausgetauschte Daten benannt (an den Pfeilen)

Umgebungs- bzw. Kontextdiagramm Beispiel Projektvorschlagssystem



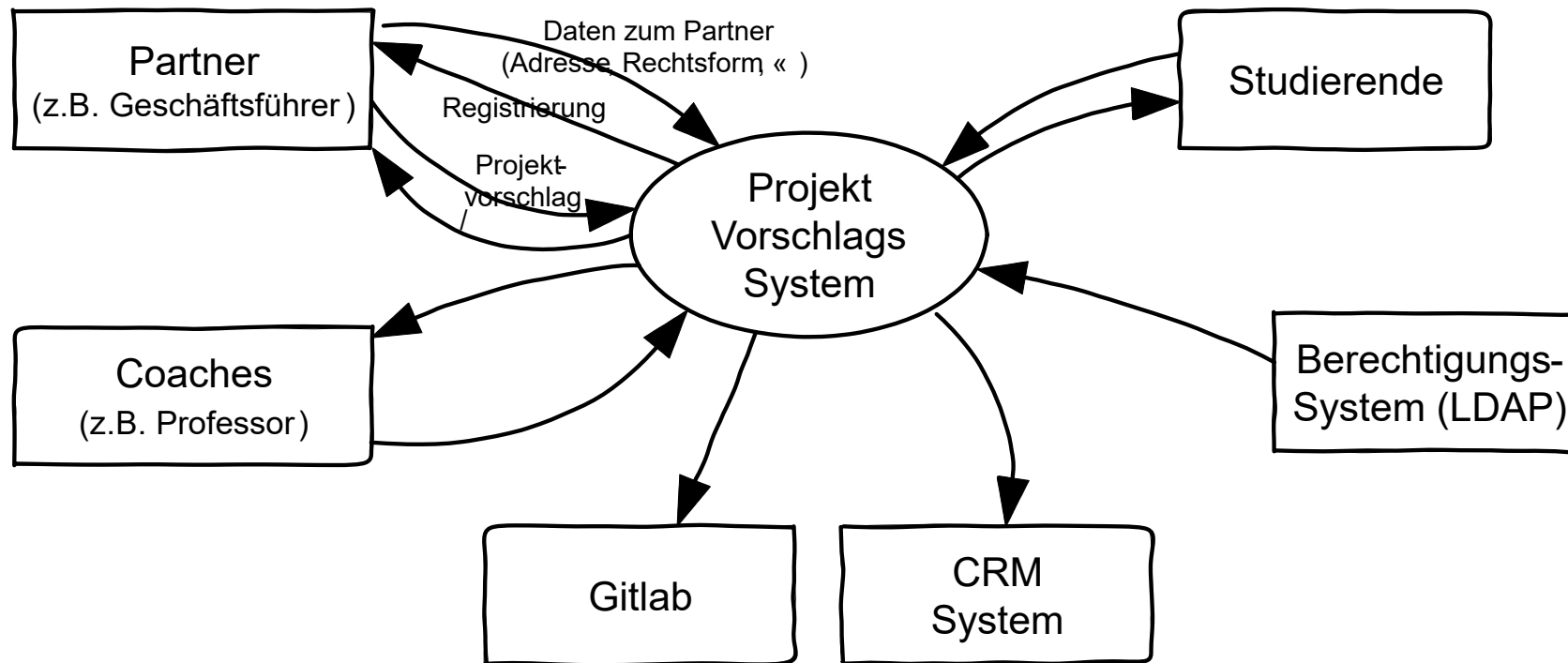
Umgebungs- bzw. Kontextdiagramm

Beispiel Projektvorschlagssystem



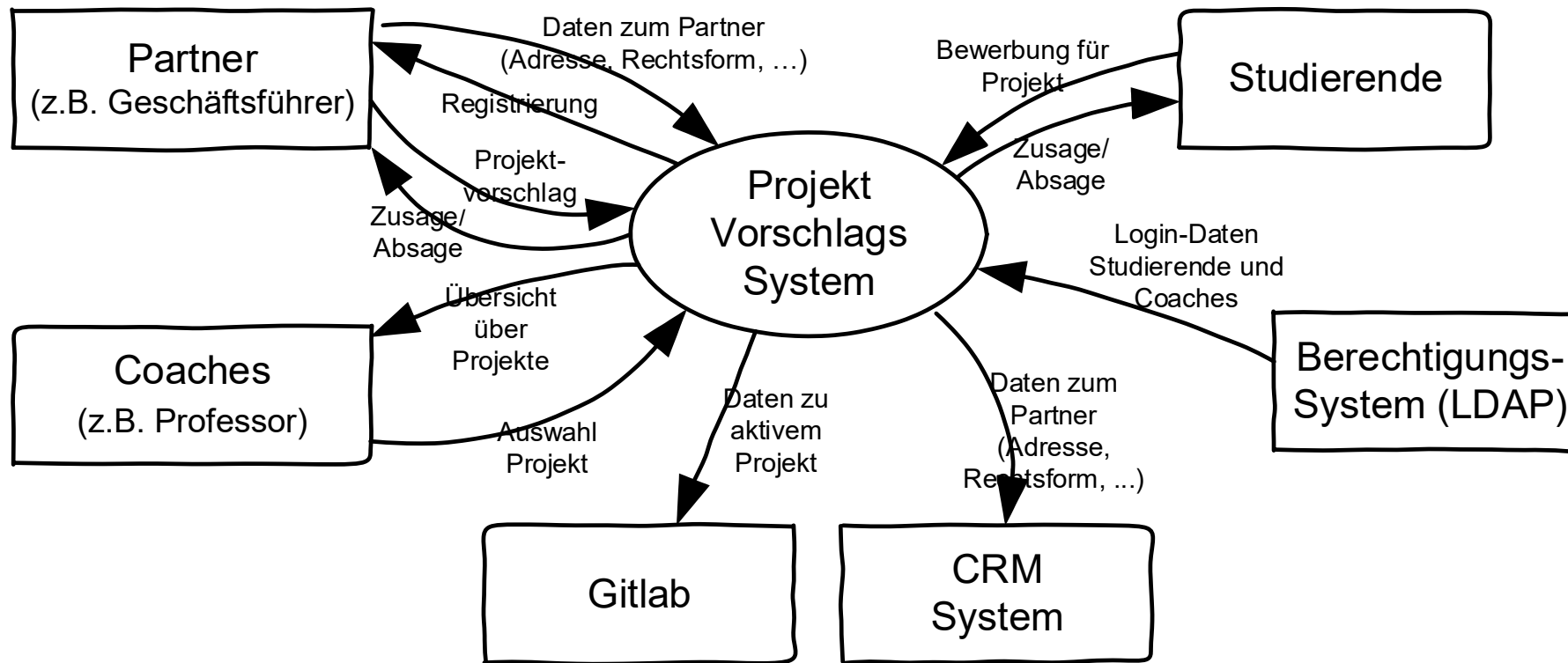
Umgebungs- bzw. Kontextdiagramm

Beispiel Projektvorschlagssystem



Umgebungs- bzw. Kontextdiagramm

Beispiel Projektvorschlagssystem



Agiles Software-Engineering

Prof. Dr. Gerd Beneken

Woche 02, Kapitel 8

Kontext verstehen - Rahmenbedingungen



Was ist eine Rahmenbedingung?

- = **Einschränkung** der Freiheit im Entwurf / beim Bau / im Vorgehen
- = Randbedingungen müssen Sie einhalten, sonst scheitert das Projekt

- Beispiele für Randbedingungen
 - Auftraggeber schreibt Prozessmodell und Liefergegenstände vor, z.B. Scrum
 - Gesetz schreibt Dokumente und Funktionen vor, z.B. DSGVO
 - Umstände schreiben Termine vor, z.B. wichtige Messe, Stichtag für Gesetzesänderung
 - Kundenportfolio schreibt Technologie vor, z.B. Smartphone mit iOS

Rahmenbedingung: Technische Umgebung

- Vorgaben durch/der Hardware, Netzwerk
 - **IT-Strategie**: Unternehmen versuchen einheitliche HW-Landschaft (nur HP, nur IBM, nur Intel-PC, ...)
 - Aktuell: ggf. strategische Vorgabe auf einen Cloud-Anbieter zu gehen
- Vorgaben durch/der Infrastruktur (Betriebssystem)
 - **Unternehmensstandards für Infrastrukturen**: nur Java / nur .NET, nur OpenSource, nur Oracle Datenbanken, nur ...
 - Infrastrukturdienste vorgeschrieben (Security, Printing, Workflow)
 - Virtualisierungs-Infrastruktur, z.B. Nur Docker mit Kubernetes, ...
 - Frontend-Technologie: z.B. Material Design, nur Flutter / Dart, nur iOS, ...
- Architekturvorgaben: Referenzarchitekturen, Produktlinien
 - **Rahmenarchitektur** und Bussysteme im Automobil stehen fest (flexray, most, can, ...)
- Standards
 - Automotive: AUTOSAR, OSEK, ...

Wirtschaftliche Rahmenbedingungen

- **Lieferzeitpunkt**
 - Entscheidung: Build-or-Buy?
z.B. Lieferzeitpunkt nur über Verwendung von Produkten (COTS) oder über Wiederverwendung zu halten.
 - Entscheidung: Big Bang oder Stufen?
z.B. System so bauen, dass es in Stufen ausgeliefert werden kann (das Wichtigste zuerst) .
- **Stückkosten** (möglichst kostengünstige Hardware), Stückzahl
 - System so bauen, dass es auf einen möglichst kleinen Mikrocontroller passt (small footprint)
- **Lebenszykluskosten**
 - Betriebskosten durch Automatisierung mindern
 - Ggf. gibt es Obergrenze für die Kosten des Betriebs auch in der Cloud
 - Infrastruktur für Software-Updates schaffen, z.B. Over-The-Air-Update
- **Investitionssicherheit**
 - Keine Infrastrukturen von Nischenanbietern
 - Vorsicht bei OpenSource Software?
 - Standards verwenden

Rahmenbedingung: Entwickelnde Organisation

- **Struktur der Organisation**
 - **Conway's Law**: Organisationen produzieren nur Systeme, die ähnlich strukturiert sind wie sie selbst
 - Vertragliche Regelungen zwischen OEM und Zulieferern (z.B. Für Motorsteuerung ist immer Bosch zuständig)
- **Verteilung des Entwicklungsteams**, Größe des Teams
 - Komponente der Architektur = Arbeitspaket für Team, Gewerk für Lieferanten
 - Verteilung der Spezialisten,
 - Arbeitskosten am Standort (Near/Offshore), ...
- **Vorangegangene Projekte** (Wissen / Erfahrungen)
 - Erfolgswahrscheinlichkeit bei ähnlichen Projekten größer
 - Lernen im Projekt -> Reibungsverlust, Lernkurve beim „Erstling“ (z.B. Ihr erstes Projekt mit Java / C# / Ruby / Python / ...)
 - Wiederverwendung der Architektur und ggf. auch des Codes
- **Entwicklungsprozess**
 - Evolutionär entstandene Architekturen unter XP vs. Entwürfe am Beginn des Projekts

Juristische Rahmenbedingungen

Beispiel DSGVO

■ Regulatorische Vorgaben

- DSGVO
- Steuerrecht
- Compliance Regeln
- ...

Workshop Privacy by Design im Software Engineering

Privacy by Design wird zwar in der europäischen Datenschutzgrundverordnung gefordert, aber über die Umsetzung in der Praxis herrscht noch große Unklarheit. Die Themenplattform [Verbraucherbelange in der Digitalisierung](#), die vom bayerischen Staatsministerium für Umwelt und Verbraucherschutz finanziert wird, führt deswegen einen Workshop zu Privacy by Design am ZD.B-Innovationslabor der HAW Rosenheim durch. Die Studierenden werden durch die Workshop im immer relevanteren Bereich „Privacy Engineering“ weitergebildet und können dieses Wissen in die spätere Arbeitspraxis einbringen.

Die Teilnehmer bekommen einen umfassenden Überblick darüber, was [Privacy by Design](#) bedeutet, wie die rechtlichen Grundlagen sind, und wie Privacy by Design in der Praxis umgesetzt werden kann und muss.

Die Zielgruppe sind Studierende und Wissenschaftler der HAW Rosenheim.



+ ZU GOOGLE KALENDER HINZUFÜGEN

Details

Datum:

17. Mai 2019

Veranstaltungskategorien:

3. Initiativen der Gründungsförderung, Innovationslabore, Verbraucherbelange

Veranstaltung-Tags:

Privacy, Privacy by Design, verbraucherbelange

Veranstaltungsort

HAW Rosenheim

Rosenheim Deutschland

Veranstalter

Themenplattform Verbraucherbelange, ZD.B