

Software Engineering I

4. Planung und Projektmanagement

Prof. Dr. Eckhard Kruse

DHBW Mannheim

Warum planen?

*"The nice thing about **not planning** is that failure comes as a complete surprise rather than being preceded by a period of worry and depression."*

Überzeugt? Oder gibt es Gegenargumente?

Planung

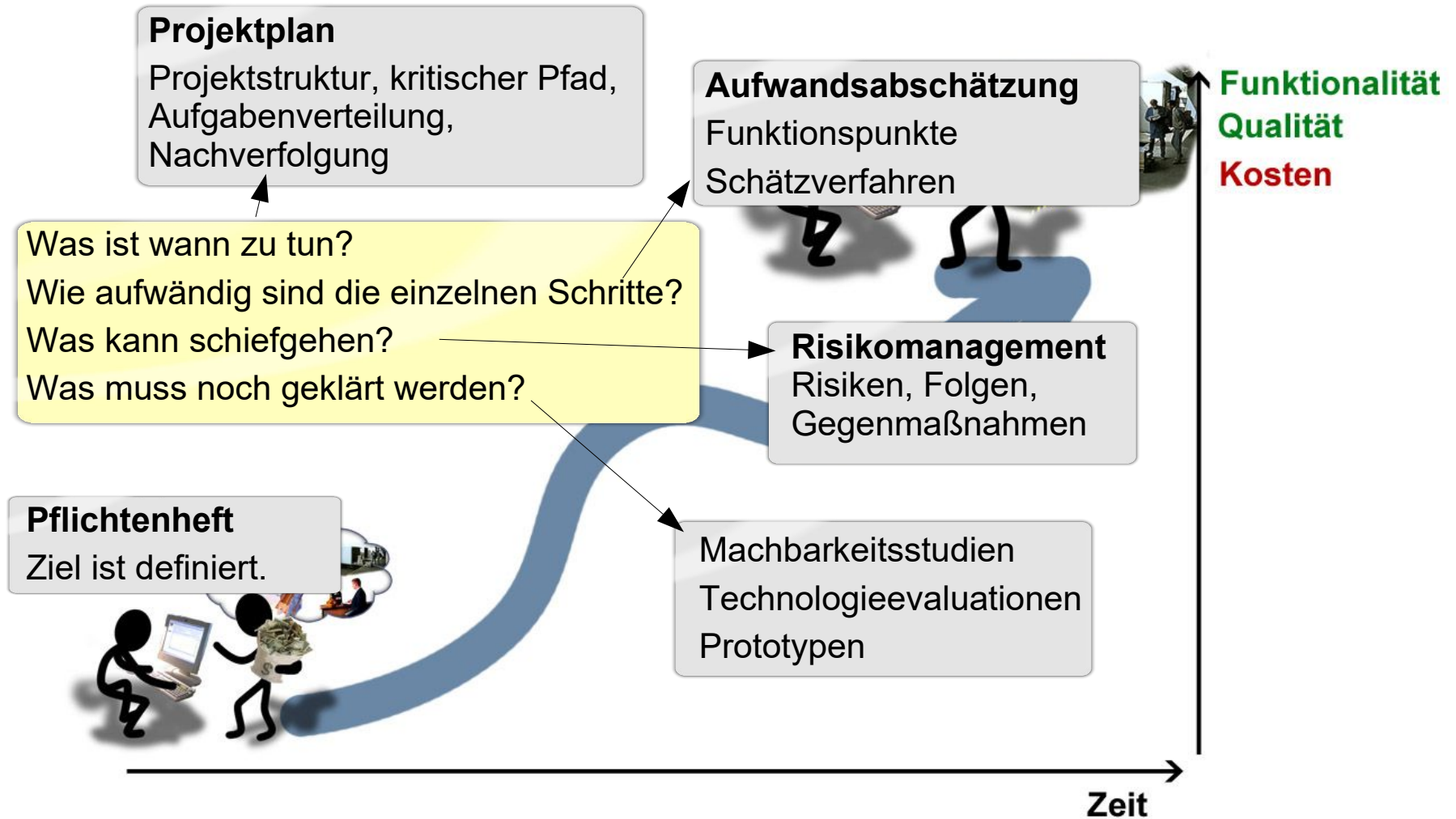
Was ist wann zu tun?
Wie aufwändig sind die einzelnen Schritte?
Was kann schiefgehen?
Was muss noch geklärt werden?

Pflichtenheft

Ziel ist definiert.

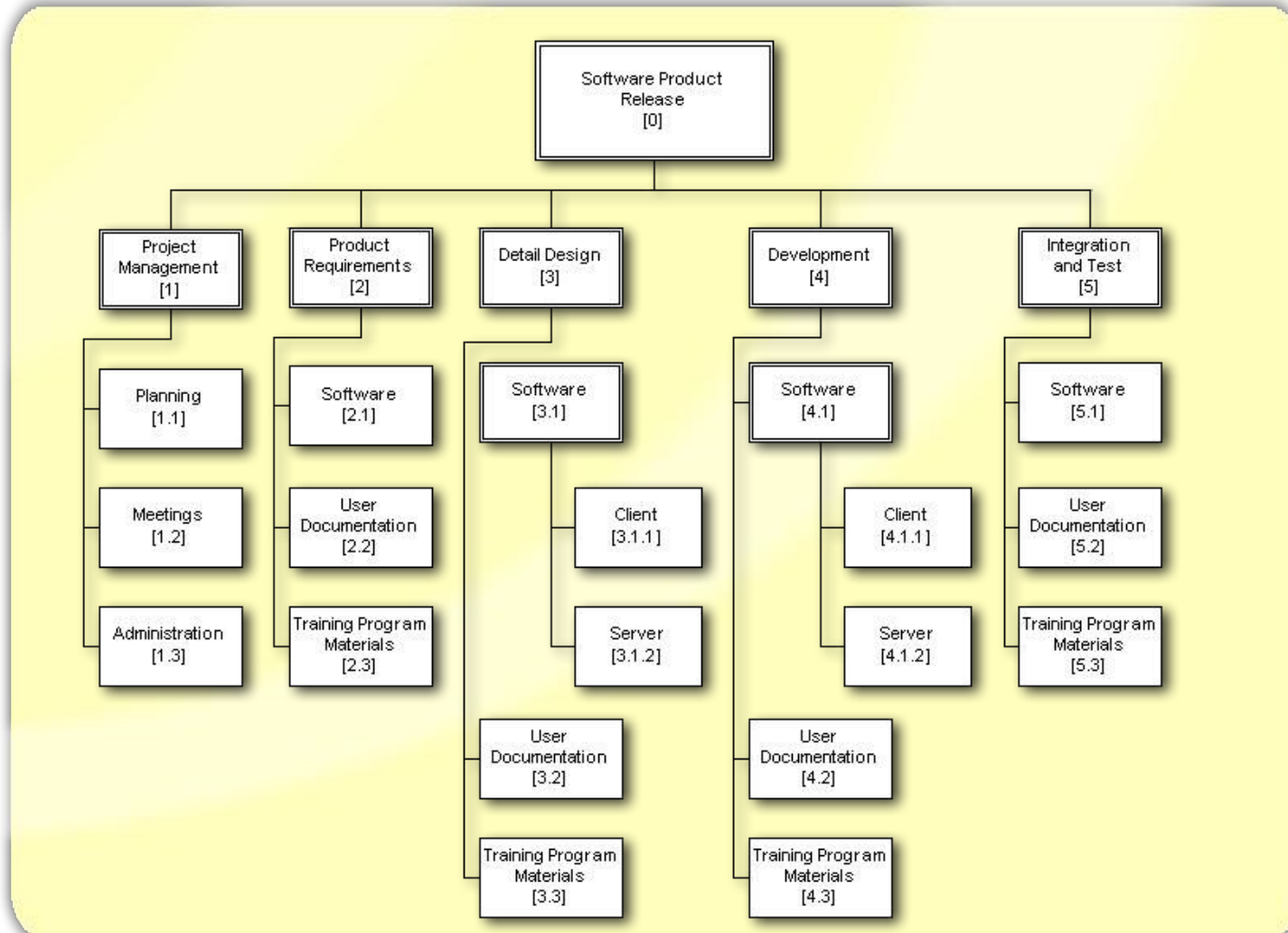


Planung



Projektstrukturplan (grob)

Beispiel

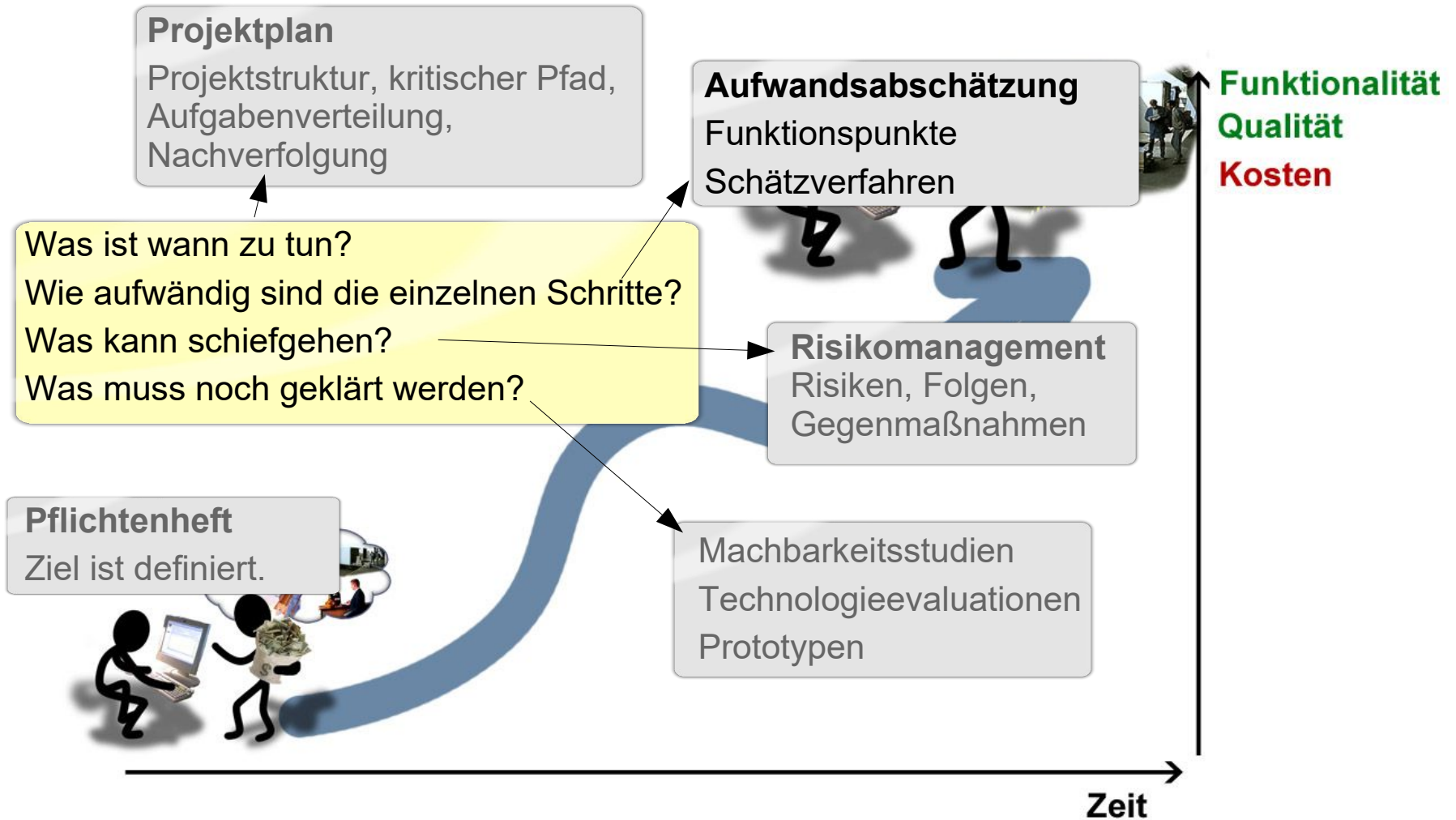


Der **Projektstrukturplan (PSP)** (*work breakdown structure, WBS*) gliedert das Projekt in überschaubare, gut planbare Arbeitspakete und beschreibt deren Abhängigkeitsbeziehungen. Er ist die Basis für die Termin- und Kostenplanung.

- Definition der Arbeitspakete: Fokus liegt auf den Ergebnissen (deliverables), nicht so sehr auf den detaillierten einzelnen Arbeitsschritten.
- Ablaufplan/Terminplan: Vorgänge (Zeiträume), Meilensteine (Zeitpunkte) und Abhängigkeiten.
- Kritischer Pfad: Sequenz der Vorgänge von Projektstart zu Projektende, bei denen es keinen Zeitpuffer gibt. D.h. eine Verzögerung eines kritischen Projektvorganges führt direkt zu einem späteren Projektende.

- **Projektplan: Template/Beispiel aus industrieller Softwareentwicklung**
- **Nachverfolgung: Gate Model**

Aufwandsabschätzung



Aufwandsschätzung

Die Kosten von Softwareentwicklungsprojekten werden zu einem Großteil durch den personellen Aufwand bestimmt.

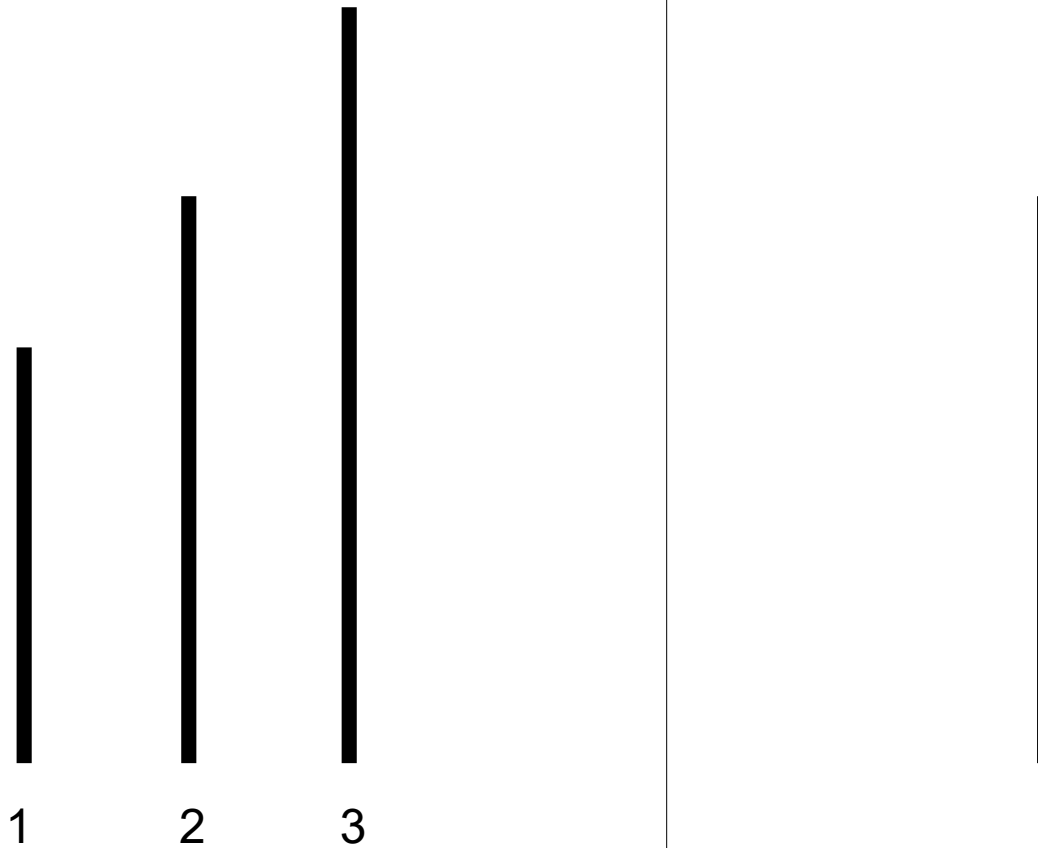
- Übliches Maß: Personentage, -wochen, -monate, -jahre.
- Personenjahre berücksichtigen Urlaub und Fehlzeiten. D.h. 1 Personenjahr ist z.B. 220 Personentage, 1650 Personenstunden (je nach Wochenarbeitszeit).

Wieviele Personentage müssen für die einzelnen Arbeitspakete geplant werden?

Vorgehen:

- Schrittweise, hierarchische Aufteilung der Arbeitspakete in Einzelaktivitäten z.B. in der Größenordnung Manntage bis max. Mannwochen
- Aufwandsschätzung für jede Einzelaktivität, z.B. mit erfahrenen Entwicklern aus dem Team
 - bester Fall, schlechterster Fall, wahrscheinlichster Fall
 - Risiken, Unsicherheiten
 - Sicherheitsmarge (da Schätzungen meist zu optimistisch)
- Aufaddieren, Durchrechnen, Zeitplanung über gesamtes Projekt, Meilensteine
- Besonderes Augenmerk auf kritischen Pfad

Schätzen Sie mal...



Solomon Asch, (1956). *Studies of independence and conformity: A minority of one against a unanimous majority*. Psychological Monographs, 70.

Aufwandsschätzung

Maße für Funktionalitätsumfang:

- Funktionspunkte (function points) = Definition "atomarer" Einzelfunktionen
- Anzahl Klassen / Komponenten, grobe Größe
- Anzahl Codezeilen (Lines of Code)

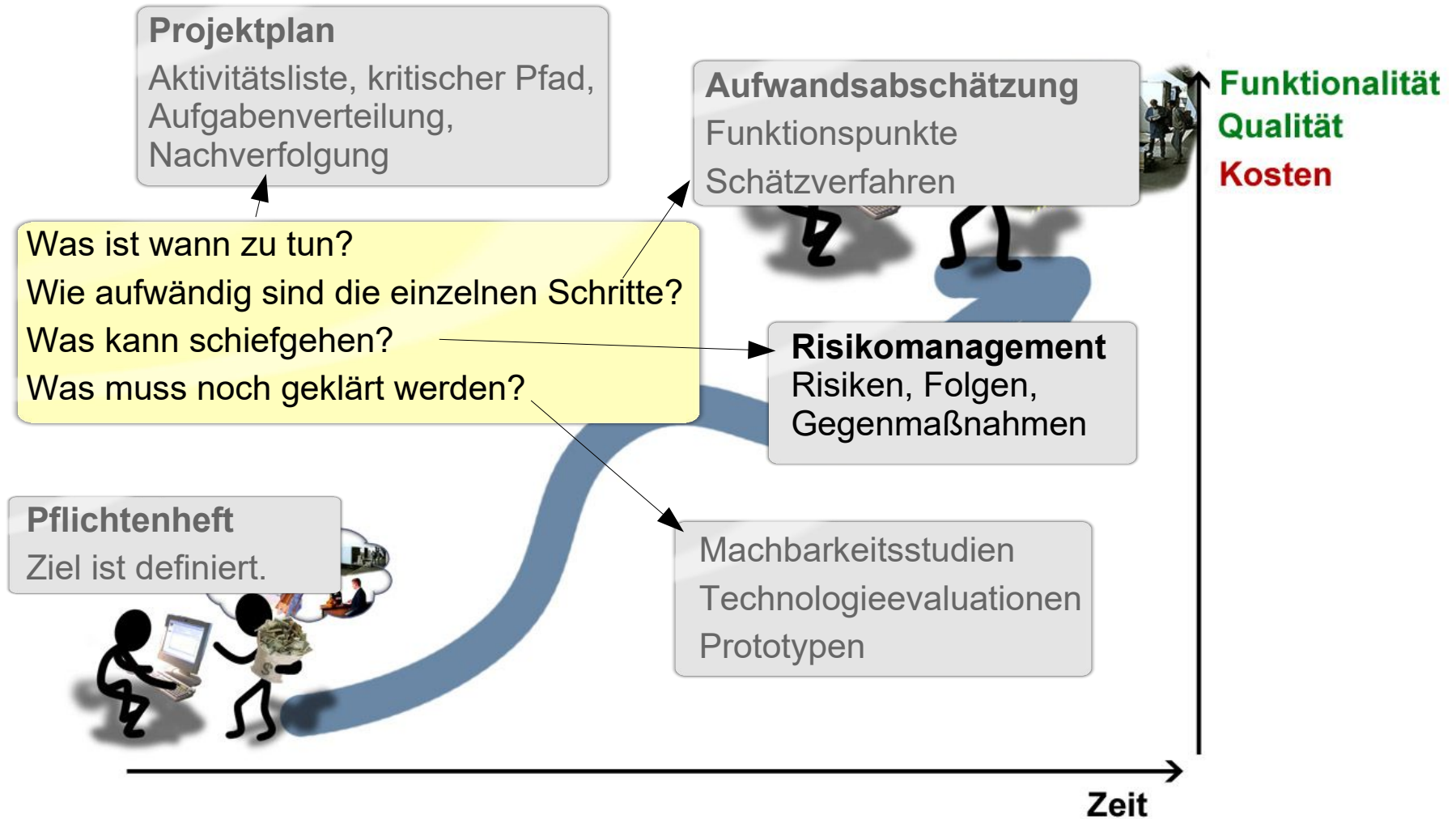
Vorsicht: Bei rein quantitativen Angaben erfolgt keine Unterscheidung zwischen einfachen und schwierigen Programmieraufgaben!

Aufwandsschätzung: Erfahrung ist entscheidend

- Vergleich mit früheren Projekten, ähnlichen Aufgabenstellungen
- Gefühl, was einfach und was aufwändig ist.
- Gefühl für Risiken, technische Probleme usw.

Dokumentierte Aufwandsschätzungen und tatsächlicher Umsetzungsaufwand können wertvolles Material für zukünftige Projekte liefern.

Risikomanagement



Risikomanagement

Risikomanagement ist der Teil des Projektmanagements, der sich mit der Identifizierung, Analyse und Beherrschung von Risiken für die geplante Projektabwicklung beschäftigt.

Vorgehen:

- Projektrisiken identifizieren und auflisten
- Mögliche Auswirkungen (auf den Projekterfolg) analysieren
- Ggf. bereits vorbeugende Maßnahmen initiieren
- Gegenmaßnahmen überlegen, falls der Risikofall eintritt
- Risikomanagement als regelmäßigen Prozessschritt durchführen, da sich Projektrisiken verändern können.



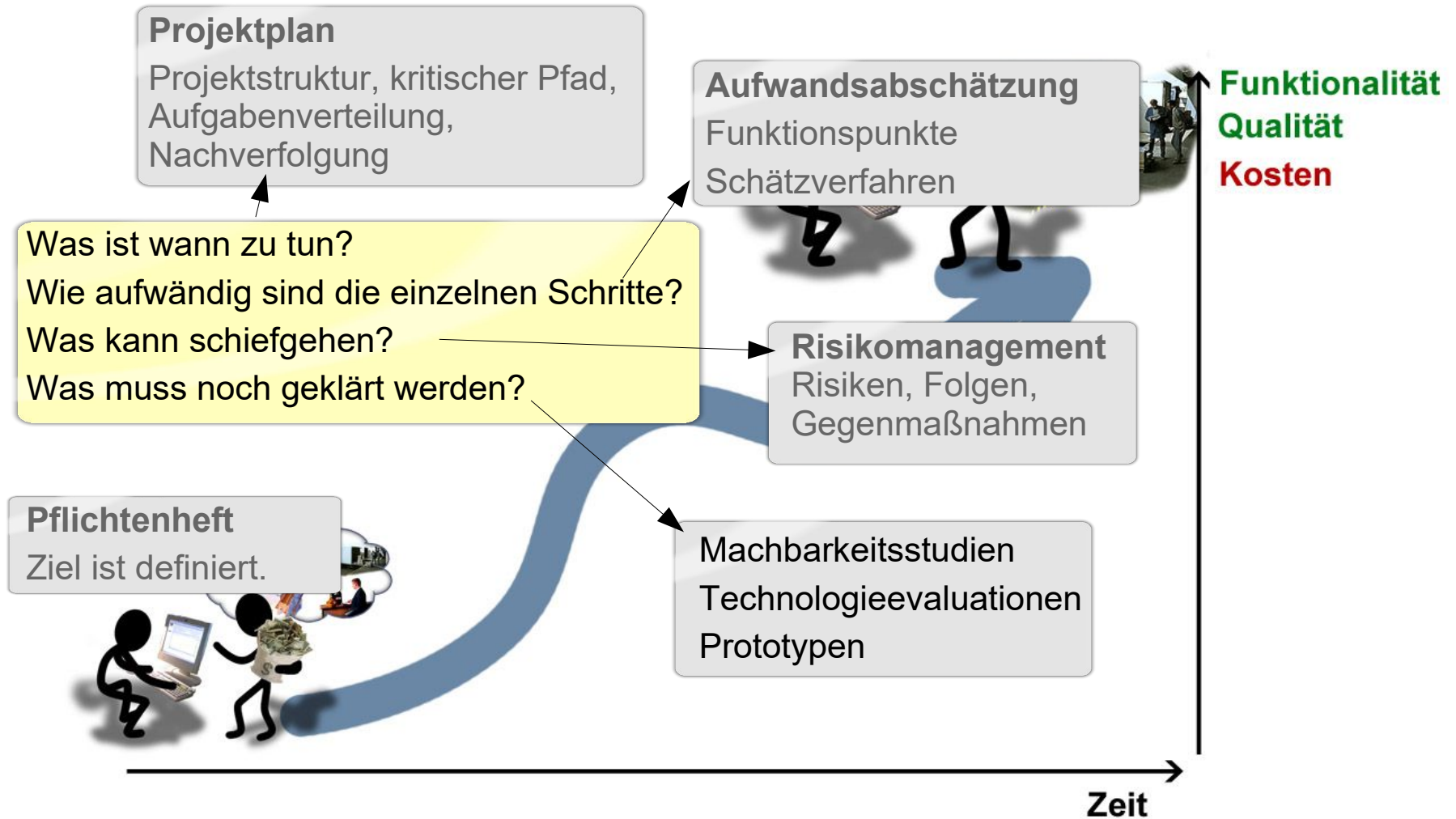
Software-Engineering-Projekt

P.10 Lieferant: Projektplanung

Auf Basis des ausgewählten Vorgehensmodells wollen Sie Ihr Projekt sorgfältig strukturieren und planen:

- a) Was sind die wesentlichen (Zwischen-)Ergebnisse/Produkte Ihres Projektes? Wann sollten diese vorliegen?
- b) Welche Aktivitäten müssen stattfinden? Welche Abhängigkeiten gibt es? (-> kritischer Pfad)
- c) Welche Rollen gibt es im Projekt, wer übernimmt welche Aufgaben?
- d) Welche Projektrisiken gibt es? Wie gehen Sie damit um?
- e) Wann und wie wollen Sie kontrollieren, ob Sie noch im Plan liegen bzw. ob Korrekturmaßnahmen erforderlich werden?
- f) Optional: Bereiten Sie eine kurze Präsentation vor, um Ihr Management zu überzeugen, dass Ihr Projekt solide geplant ist und ein Erfolg wird.

Machbarkeitsstudien



Machbarkeitsstudie

Um Fehlinvestitionen zu verhindern und Risiken zu minimieren, kann bei Zweifeln an dem Erfolg eines geplanten Projektes eine **Machbarkeitsstudie** vorangestellt werden.

Eine Machbarkeitsstudie soll Fragen zum geplanten Projekt klären, z.B. in Bezug auf:

- Wirtschaftlichkeit
- Technische Machbarkeit
- Marktsituation, Unternehmens- und Produktstrategie
- Patentrechtliche Fragen
- Aufwandsabschätzung
- Make or Buy Entscheidungen
- Grundlage für Angebotserstellungen

1 General Information

- 1.1 Project Administration
- 1.2 Project Goal
- 1.3 Project Result

2 Business

- 2.1 Customer issue
- 2.2 Solution
- 2.3 Benefits
 - 2.3.1 End-Customer Benefit
 - 2.3.2 Company Benefit
- 2.4 Targeted Customers
- 2.5 Market
- 2.6 Business Impact
- 2.7 Relation to Product Portfolio
- 2.8 Competition Evaluation
- 2.9 Intellectual Property
 - 2.9.1 Patent Search: Results of Interest
 - 2.9.2 Potential New Intellectual Property

3 Technology

- 3.1 User Operations
- 3.2 Implementation Approach
- 3.3 Project Main Requirements
- 3.4 Other Alternatives

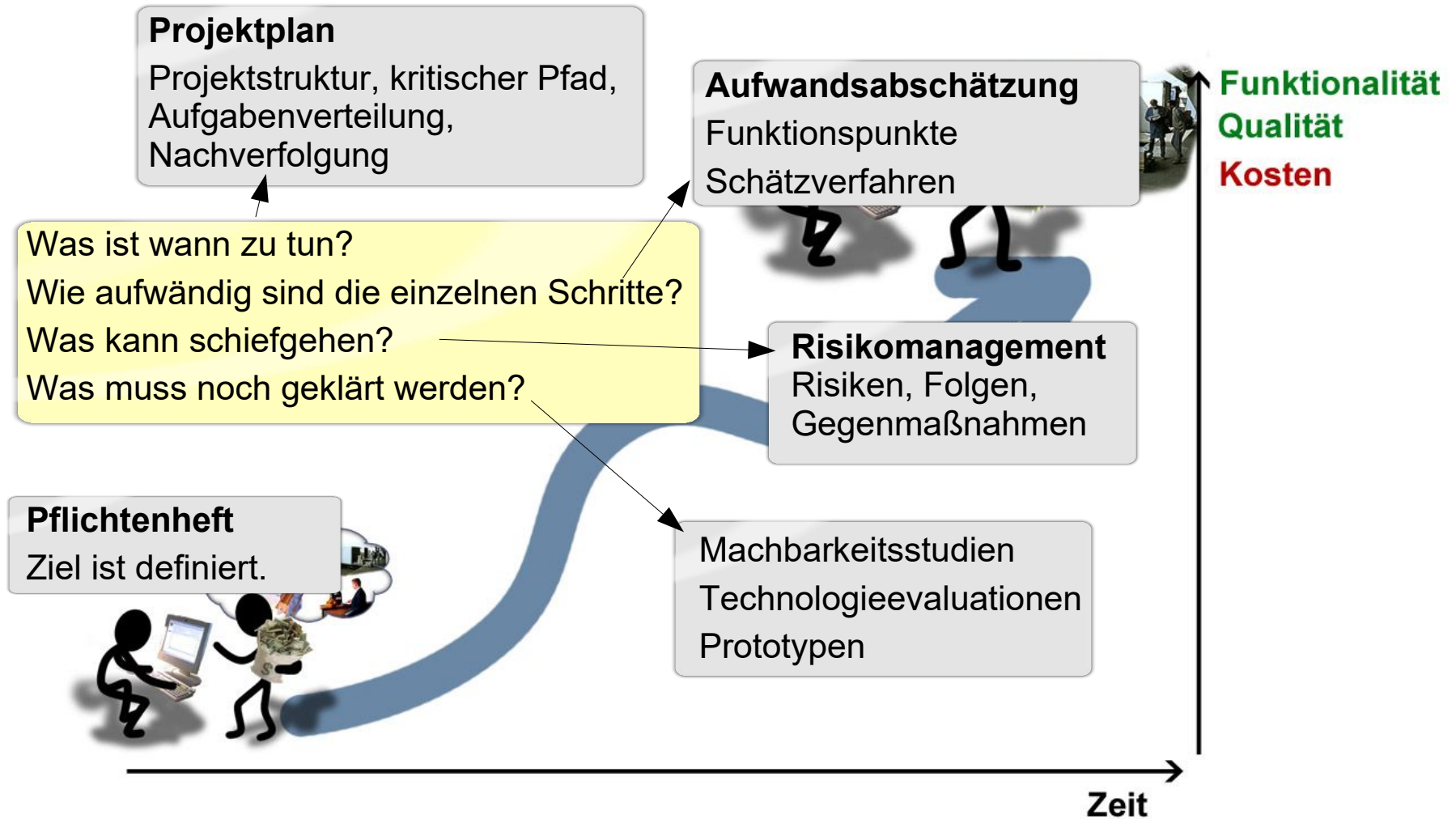
4 Project Management

- 4.1 Target Dates
- 4.2 Estimated Cost
- 4.3 Project Organization
- 4.4 End Customer contacts

5 Risks

- 6 References
- Change History
- Review History

Planung



Prototyp

Ein **Prototyp** ist die rudimentäre, ansatzweise Implementierung (von Teilen) des zu entwickelnden Softwareprodukts. Im Vordergrund bei der Prototypentwicklung steht, möglichst schnell einen ersten Eindruck des Produktes zu erhalten - ohne Anspruch auf Vollständigkeit, hohe Software-Qualität oder gute Softwarearchitektur.

Einsatzbereiche eines Prototyps (insbesondere in den frühen Projektphasen):

- Demonstration (z.B. der Benutzerschnittstelle) bei den Interessengruppen des Projektes
- Ausprobieren neuer Ideen – z.B. im Rahmen der Anforderungsanalyse
- Aufwandsabschätzung, wie komplex bestimmte Entwicklungsaufgaben werden können
- Evaluation verschiedener technischer Lösungsansätze

Die eigentliche Produktentwicklung sollte (in aller Regel) nicht auf Basis des Prototyp-Codes erfolgen, sondern mit einem neuen, sauberen Entwurf!

Prototypen - Begriffe

- Ein **Demonstrationsprototyp** vermittelt einen ersten, sehr groben Eindruck von dem Produkt.
- Ein **horizontaler Prototyp** betrachtet eine komplette Systemschicht (z.B. die Benutzeroberfläche) ohne die abhängige Funktionalität in anderen Systemschichten (z.B. ohne Business Logic).
- Ein **vertikaler Prototyp** realisiert eine ausgewählte Funktion des Systems durch alle Schichten (z.B. Benutzeranmeldung: Betrifft UI, Administration, Datenhaltung, IT-Sicherheit usw.)
- Ein **explorativer Prototyp** ist eine Spielwiese, um neue Ideen auszuprobieren und in der Anforderungsanalyse offene Punkte zu untersuchen.
- Ein **experimenteller Prototyp** dient zur Untersuchung technischer Realisierungsmöglichkeiten und soll den Nachweis erbringen, dass konzipierte Lösungsvorschläge in der Praxis auch anwendbar sind.
- Der **evolutionäre Prototyp** ist im Gegensatz zu den anderen Prototypen kein Wegwerfprodukt, sondern soll stufenweise bis zum fertigen Produkt verfeinert und weiterentwickelt werden.

Mock-up

Ein **Mock-up** ist ein Vorführmuster, um das Aussehen des zu entwickelnden Produktes zu demonstrieren (allerdings ohne Funktionalität).

- Einsatzbereich insbesondere in der Anforderungsanalyse, um Kunden/Anwender-Feedback zu erhalten.
- Mock-ups von Softwareprodukten stellen die Benutzerschnittstelle dar (= den sichtbaren Teil der Software) und können z.B. einfach mit Zeichenprogrammen o.ä. realisiert werden.
- Ein Mock-up ist kein Prototyp (denn ein Prototyp umfasst immer auch tatsächlich implementierte Funktionalität).



Software-Engineering-Projekt

P.11 Lieferant: Use Cases und Mock-up

Um sicherzugehen, dass Sie die Anforderungen Ihres Kunden richtig erfasst haben, bereiten Sie Use Cases (Anwendungsfälle) und ein Mock-up („Vorführmodell“) vor:

- a) Charakterisieren Sie typische Anwendungsszenarien des Systems.
- b) Entwerfen Sie das Aussehen der Benutzeroberfläche, z.B. mit einem Malprogramm. Was sind wesentliche grafische Elemente, welches sind die Bedienelemente, Buttons usw.?



Software-Engineering-Projekt



P.12 Kunde-Lieferant: Präsentation Mock-up

Der Lieferant stellt dem Kunden seine Entwürfe vor. Ist das Projekt auf dem richtigen Weg oder gibt es bei den Anforderungen noch Missverständnisse, die geklärt werden müssen?