

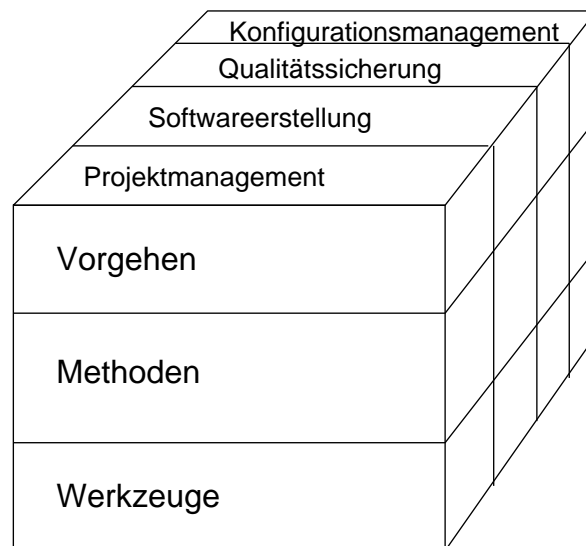
Das V-Modell

Das V-Modell ist Bestandteil des Standardisierungskonzepts der Bundesbehörden. Dieses Konzept hat folgende Eigenschaften:

- verbindlich für IT-Vorhaben in Bundesbehörden (wird jedoch auch in der Wirtschaft eingesetzt)
- deckt die Softwareentwicklung, das Projektmanagement, die Qualitätssicherung und das Konfigurationsmanagement ab
- beschreibt die Vorgehensweise (*V-Modell*), die anzuwendenden Methoden und die einzusetzenden Werkzeuge

Slide 1

Struktur des Standardisierungskonzepts



Slide 2

Eigenschaften des V-Modells

Das V-Modell

- **beschreibt** *Aktivitäten* und *Produkte*
- **regelt** die Gesamtheit der Aktivitäten und Produkte sowie die *Produktzustände* und die *Abhängigkeiten* zwischen Aktivitäten und Produkten

Slide 3

Wichtig: das V-Modell beschreibt den Softwareentwicklungsprozeß nur aus funktionaler Sicht, d.h. es werden keine Aussagen über die konkrete organisatorische Umsetzung getroffen. Die Aufteilung von Aufgaben auf Bearbeiter bzw. Organisationseinheiten wird vor Projektdurchführung festgelegt.

Erzeugnisstruktur

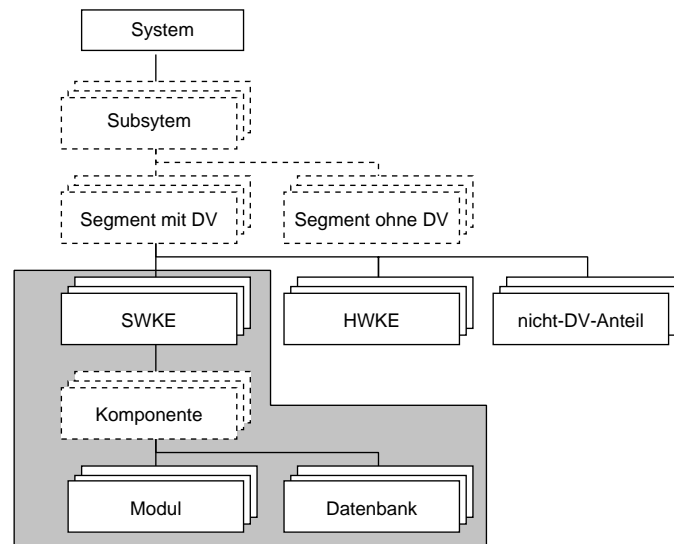
Das Gesamtsystem (inklusive HW und SW) wird in unterschiedliche Bestandteile aufgelöst:

- Subsysteme
- Segmente (mit DV-Anteil/ohne DV-Anteil)
- Konfigurationseinheiten (SW, HW und ohne DV-Anteil)
- Komponenten
- Module/Datenbanken

Slide 4

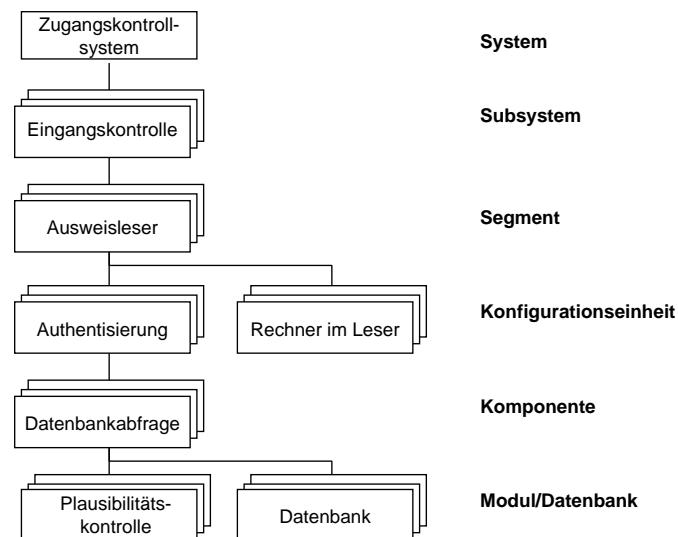
Erzeugnisstruktur

Slide 5



Beispiel: Zugangskontrollsystem

Slide 6



Struktur und Inhalt des V-Modells

Das V-Modell besteht aus folgenden Elementen:

- Aktivitäten
- Produkte
- Beschreibungsmuster
- Submodelle

Slide 7

Aktivitäten und Produkte

Aktivitäten: Tätigkeiten, die hinsichtlich ihres Ergebnisses und ihrer Durchführung genau beschrieben werden können. Sie sind durch Teilaktivitäten verfeinerbar, die Zwischenergebnisse erzeugen. Aktivitäten der obersten Ebene werden *Hauptaktivitäten* genannt.

Slide 8

Produkte: Bearbeitungsgegenstand oder Ergebnis einer Aktivität, kann wiederum auch in Teilprodukte zerlegt werden. Produkte können die Zustände *geplant*, *in Bearbeitung*, *vorgelegt* und *akzeptiert* annehmen.

Notation:



Produktzustände

geplant: Eingangszustand aller Produkte

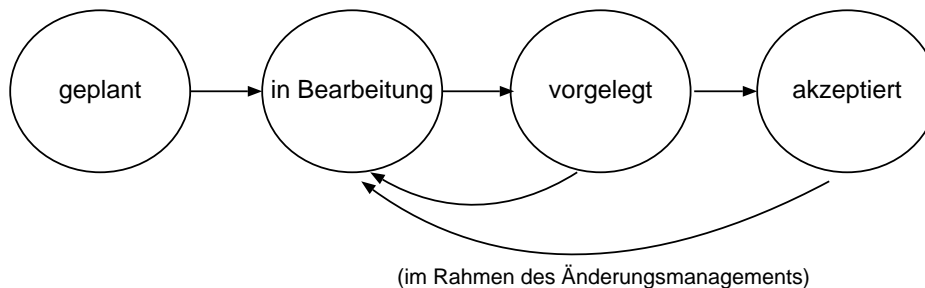
in Bearbeitung: Produkt befindet sich entweder beim Entwickler oder in der Produktbibliothek

Slide 9 **vorgelegt:** das Produkt ist aus der Sicht des Entwicklers fertig und befindet sich in der Konfigurationsverwaltung. Es wird QS-Prüfung unterzogen und entweder zurückgewiesen oder akzeptiert

akzeptiert: das Produkt ist freigegeben und darf nur in einer neuen Version geändert werden

Produktzustände und Übergänge

Slide 10



Beschreibungsmuster

Aktivitäten, deren Zerlegungen und Produkte werden in Beschreibungsmustern festgelegt. Diese bestehen aus:

Slide 11

- Aktivitätenschema (Produktfluß und Abwicklung)
- graphische Aktivitätenzerlegung (Teilaktivitäten, Abhängigkeiten, Teilergebnisse)
- Produktschema (kurze inhaltliche Beschreibung, Gliederung)

Produktfluß und Abwicklung

Produktflüsse beschreiben

die Eingangs- und Ausgangsprodukte für eine bestimmte Aktivität in einer Tabelle:

Slide 12

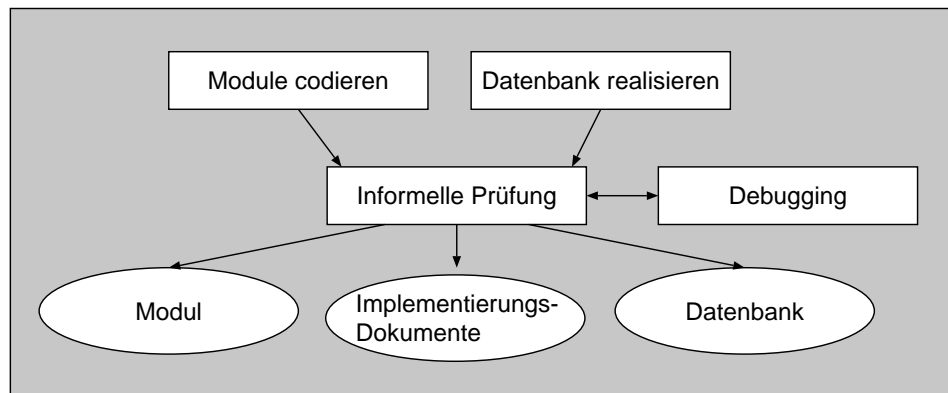
von		Produkt	nach		QS	KM
Aktivität	Zustand		Aktivität	Zustand		
SWE 2.5	akzeptiert	DV-Architektur	-	-		
SWE 3.6	akzeptiert	SW-Anforderungen	-	-		
-	-	SW-Architektur	SWE 4.2	in Bearb.		

Die **Abwicklung** beschreibt in Textform wie die Aktivität durchzuführen ist. Zusätzlich können *Empfehlungen* und *Erläuterungen* angegeben werden.

Aktivitätenzerlegung

Die Aktivitätszerlegung wird bei der Verfeinerung von Aktivitäten angewendet. Sie beschreibt den Zusammenhang von Teilaktivitäten und -produkten in graphischer Form (hier die Aktivität SWE 6/Implementierung):

Slide 13



Die Submodelle

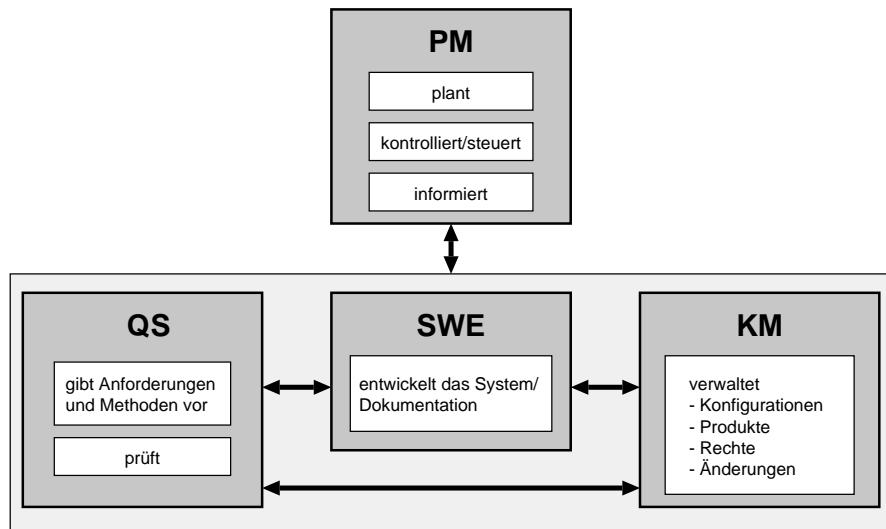
Die Aktivitäten der Softwareentwicklung werden auf vier Submodelle verteilt:

- Softwareerstellung (**SWE**)
- Qualitätssicherung (**QS**)
- Konfigurationsmanagement (**KM**)
- Projektmanagement (**PM**)

Slide 14

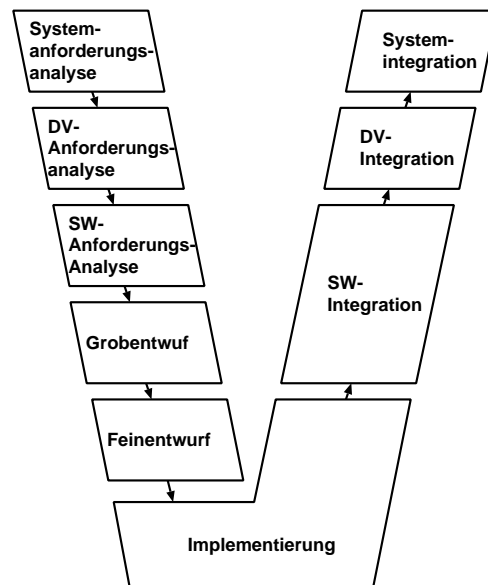
Zusammenspiel der Submodelle

Slide 15



Submodell "Softwareerstellung" (SWE)

Slide 16



Systemanforderungsanalyse und -entwurf (SWE 1)

betrifft das Gesamtsystem und besteht aus folgenden Aktivitäten:

Ist-Analyse: Datenbestände, DV-Ausrüstung, Zeit/Mengengerüst, Schwachstellen

Slide 17

Bedrohungs- und Risikoanalyse: zu schützende Objekte, Eintrittswahrscheinlichkeiten

Systemanforderungen definieren: Anforderungen an Funktionalitäten, Daten, Qualitätsmerkmale aus Anwendersicht

System fachlich strukturieren: Funktions- und Datenstruktur aus Anwendersicht

Systemanforderungsanalyse und -entwurf (SWE 1)

System technisch entwerfen: Aufteilung in Subsysteme, Segmente oder Softwarekonfigurationseinheiten

Machbarkeitsanalyse: Verwendung von Fertigprodukten, Wirtschaftlichkeit, Tolerierbarkeit von Restrisiko

Slide 18

Systemanforderungen zuordnen: Zuordnung der Anforderungen zu den Systembestandteilen

Systemintegration planen: Planen des Zusammenführens der Systembestandteile, Planen der Inbetriebnahme

Wichtige Produkte in SWE 1 sind die *Systemanforderungen*, die *Systemarchitektur* und der *Systemintegrationsplan*

DV-Anforderungsanalyse und -entwurf (SWE 2)

besteht aus folgenden Aktivitäten:

- DV-Ist-Analyse
- DV-Anforderungen analysieren
- Machbarkeitsanalyse
- DV-Segment entwerfen
- DV-Anforderungen zuordnen
- DV-Integration planen

Slide 19

DV-Anforderungsanalyse und -entwurf (SWE 2)

Die Aktivitäten korrespondieren mit den Aktivitäten aus SWE 1, jedoch betreffen sie hier nur einzelne DV-Segmente.

Wichtige Produkte sind die *DV-Anforderungen*, die *DV-Architektur*, der *Schnittstellenentwurf* und der *DV-Integrationsplan*

Slide 20

SW-Anforderungsanalyse (SWE 3)

behandelt einzelne SWKE und besteht aus folgenden Aktivitäten:

- Ist-Analyse
- Allgemeine Anforderungen definieren
- Anforderungen an externe Schnittstellen definieren
- Anforderungen an Funktionen und Daten definieren
- Qualitätsanforderungen definieren
- Anforderungen an Entwicklungsumgebung und Zielrechner definieren

Slide 21

Als Produkt werden die *SW-Anforderungen* erstellt.

Grobentwurf (SWE 4)

Zerlegt die SWKE in einzelne Prozesse und definiert die Kommunikation und Synchronisation zwischen den Prozessen. Weiter werden die SWKE dann in Module aufgeteilt. Aktivitäten sind:

- SW-Architektur entwerfen
- Prozesse entwerfen
- SWKE-interne Schnittstellen entwerfen
- SWKE-Integration definieren

Slide 22

Als Ergebnis wird die *SW-Architektur*, der *Schnittstellenentwurf* und der *SWKE-Integrationsplan* erstellt.

Feinentwurf (SWE 5)

Entwurf der Realisierung der Module und Datenbanken:

- Komponente/Modul/Datenbank beschreiben
- Betriebsmittel- und Zeitbedarf analysieren

Slide 23

Hieraus resultieren der *Datenkatalog* und der *SW-Entwurf*.

Implementierung (SWE 6)

Umsetzung der Module und Datenbanken:

- Module codieren
- Datenbanken realisieren
- Informelle Prüfung
- Debugging

Slide 24

Produkte sind *Module* und *Datenbanken*.

SW-Integration (SWE 7)

Zusammenführung der Module und Datenbanken zu SWKE:

- Zu SWKE-Teilstruktur integrieren
- Komponente informell prüfen

Slide 25

- Zur SWKE integrieren
- SWKE informell prüfen

Ergebnis der SW-Integration sind die *SWKE*.

DV-Integration (SWE 8)

Zusammenführung von SWKE und HWKE zu DV-Segmenten:

- Zum DV-Segment integrieren
 - Tuning
- Slide 26**
- Beitrag zur Einführungsunterstützung leisten
 - DV-Segment informell prüfen

Die DV-Integration liefert *vollintegrierte* und *teilintegrierte DV-Segmente*

Systemintegration (SWE 9)

Integration des Gesamtsystems aus den Subsystemen oder DV-Segmenten:

- Zum Subsystem integrieren
- Subsystem informell prüfen
- Zum System integrieren
- System informell prüfen

Slide 27

Produkte im Submodell SWE

Produkte, die bei der SWE erstellt werden:

Systemprodukte: Systemanforderungen, Systemarchitektur

DV-Produkte: DV-Anforderungen, DV-Architektur

Slide 28 **SW-Produkte:** SW-Anforderungen, SW-Architektur, SW-Entwurf,
Implementierungsdokumente

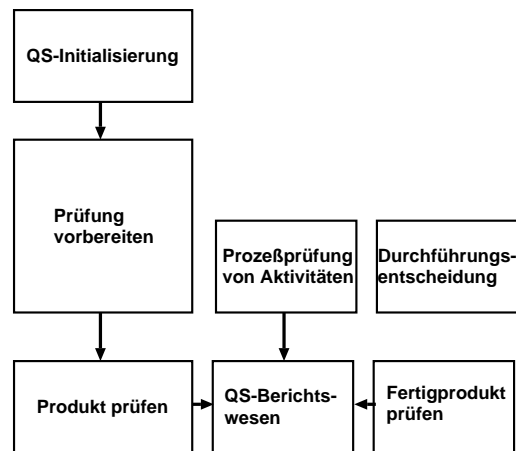
Integrationsdokumente: Integrationsplan

Handbuchinformation: Anwenderhandbuch, Diagnosehandbuch,
Betriebshandbuch

Begleitende Dokumente: Datenkatalog, Schnittstellenentwurf

Submodell "Qualitätssicherung" (QS)

Slide 29



Aktivitäten im Submodell QS

Folgende Aktivitäten werden im Submodell QS durchgeführt:

QS-Initialisierung: legt den organisatorischen Rahmen im QS-Plan und in Prüfplänen fest

Slide 30

Prozeßprüfung von Aktivitäten: Prüft die Einhaltung bestimmter Vorgehensweisen

Produktprüfung vorbereiten: Erstellen von Prüfspezifikationen und -prozeduren, Erfolg und Mißerfolg einer Prüfung muß eindeutig und nachvollziehbar sein

Aktivitäten im Submodell QS

Produkt prüfen: Prüfen nach *formalen* und nach *inhaltlichen* Kriterien gemäß den Prüfspezifikationen

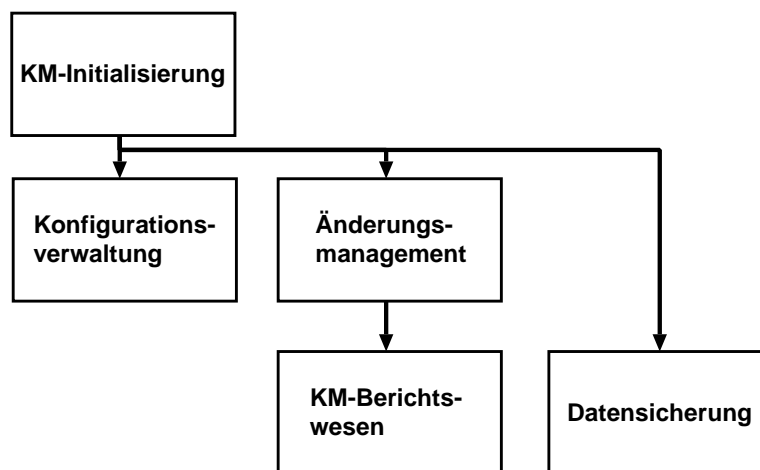
Durchführungsentscheidung: Entscheidung, ob die nächste SWE-Hauptaktivität durchgeführt werden darf

Slide 31

Fertigprodukt prüfen: Prüft die Qualitätsanforderungen an das fertige Produkt

QS-Berichtswesen: Auswertung von Prüfprotokollen und Übergabe an das Projektmanagement

Submodell “Konfigurationsmanagement” (KM)



Slide 32

Aktivitäten im Submodell KM

KM-Initialisierung: legt organisatorischen Rahmen für die Konfigurationsverwaltung sowie alle Einsatzmittel fest

Konfigurationsverwaltung: Verwaltung von Produkten, Konfigurationen und Rechten, Festlegung einer zentralen Datenadministration

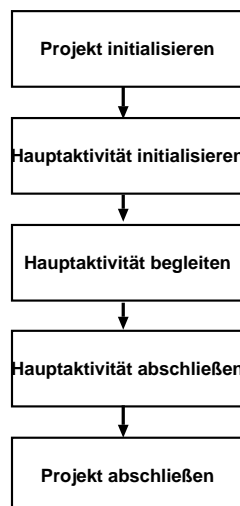
Slide 33

Änderungsmanagement: Erfassung von Fehlermeldungen, Problemmeldungen, Verbesserungsvorschlägen etc.

Datensicherung

KM-Berichtswesen: Zur Vorbereitung von Durchführungsentscheidungen und zur Information des Projektmanagements

Submodell "Projektmanagement" (PM)



Slide 34

Aktivitäten im Submodell PM

Projekt initialisieren: regelt den organisatorischen Rahmen für das gesamte Projekt im *Projektplan* und *Projekthandbuch*. Legt Modalitäten für die projektinterne Zusammenarbeit und die Schnittstelle zum Auftraggeber fest. Anpassung des Vorgehensmodells (*Tailoring*).

Slide 35

Projekt begleiten: Unterteil in die Phasen *Initialisieren*, *Begleiten* und *Abschließen* der einzelnen Aktivitäten im Projekt.

Projekt abschließen: Aufbereitung der Ergebnisse, Soll-Ist-Vergleich

Rollen im V-Modell

Das V-Modell legt *Rollen* fest, die durch Erfahrungen, Kenntnisse und Fähigkeiten definiert sind. Diese Rollen werden dann im Projektablauf den Mitarbeitern zugewiesen:

Slide 36

SWE-Rollen: Systemanalytiker, Systemdesigner, DV-Analytiker, DV-Designer, SW-Analytiker, SW-Designer, Programmierer, Supportberater, Applikationsberater, HW-Berater, Technischer Autor

PM-Rollen: Projektmanager, Projektleiter, Projektassistent

QS-Rollen: QS-Manager, QS-Verantwortlicher, Qualitätsprüfer, QS-Assistent

KM-Rollen: KM-Leiter, Konfigurationsadministrator, Datenschutz- und Datensicherheitsbeauftragter

Rollen im V-Modell

Den einzelnen Rollen werden detailliert Aufgaben zugeordnet (*ausführend, beratend, mitwirkend*). Die Zuordnung von Rollen zu Mitarbeitern kann frei erfolgen mit zwei Ausnahmen:

Slide 37

- Personen mit QS-Rollen dürfen nicht an der Erstellung der von ihnen zu prüfenden Produkte beteiligt sein (dies ist abhängig von der Kritikalität unterschiedlich streng zu handhaben)
- Zu trennen ist auch die Rolle des Projektmanagers von QS- und SWE-Rollen

Es ist somit eine Mindestanzahl von zwei bis drei Personen an einem Projekt beteiligt.

Tailoring

Das V-Modell soll sowohl für kleine als auch sehr große Projekte geeignet sein. Demzufolge definiert es alle Aspekte, die in diesen Projekten auftreten können. Das *Tailoring* sieht vor, aus dem V-Modell durch Streichungen ein an das Projekt angepaßtes Modell zu erstellen. Es sind jedoch auch schon Beispielmuster (inklusive Dokumente) erhältlich.

Slide 38

Zusammenfassung

- Das V-Modell ist ein übergreifendes Modell zur Abwicklung von Softwareentwicklungsvorhaben
- Es muß vor dem Einsatz angepaßt werden

Slide 39

- Wegen der hohen Zahl und Komplexität von zu erstellenden Produkten ist eine Werkzeugunterstützung fast zwingend
- Es verlangt erhöhte Kooperation zwischen Auftraggeber und Auftragnehmer, kann dabei aber Kommunikationsprobleme verringern