



Vorlesung Softwaretechnik I (SS 2024)

2. Projekte, Personen, Prozesse, Produkte

Prof. Dr. Jens Grabowski

Tel. 39 172022

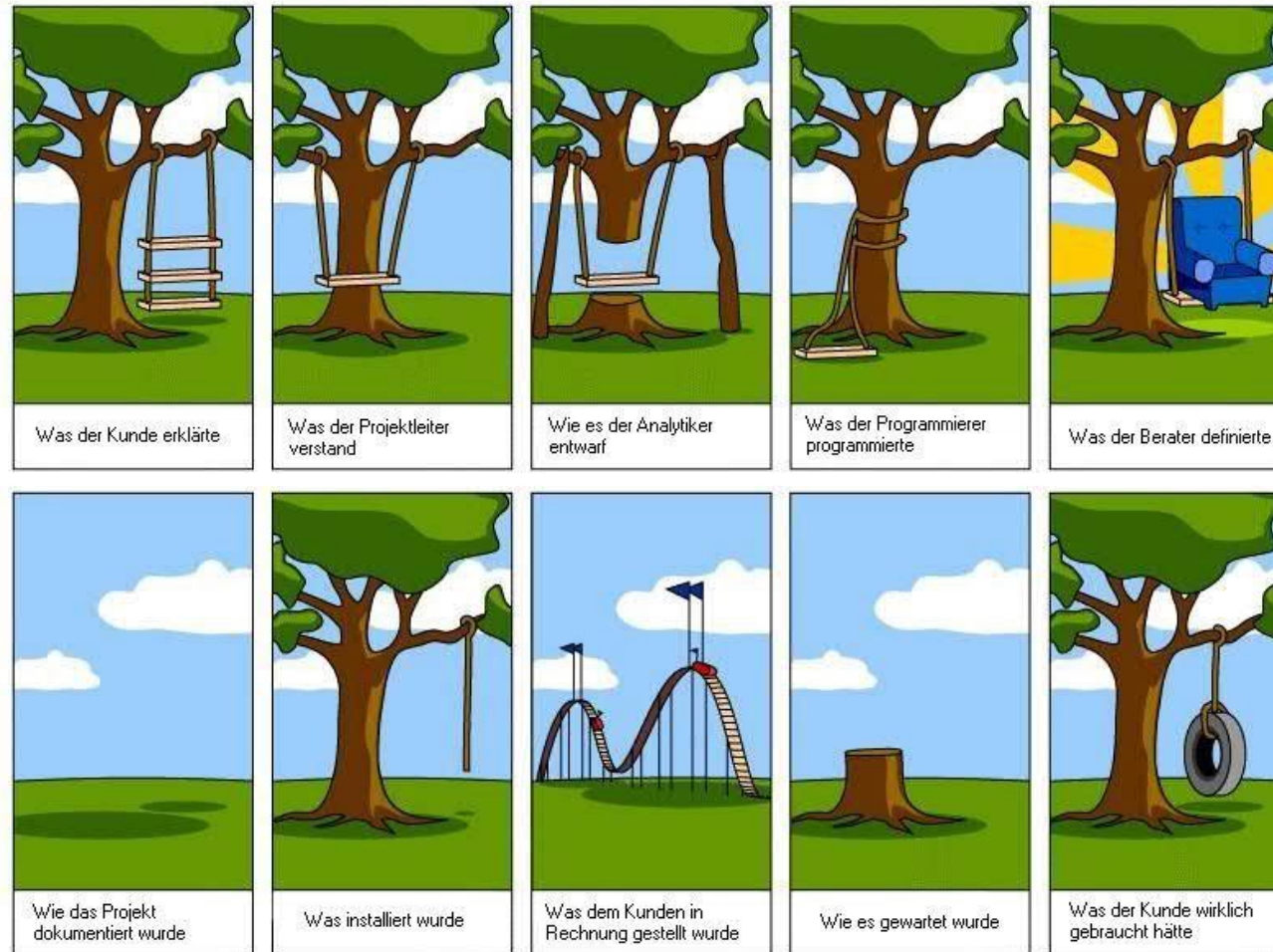
grabowski@informatik.uni-goettingen.de



Inhalt

- Einführung
- Das Projekt
- Personen
- Prozesse
- Produkte
- Lernziele

Einführung – Typisches Softwareprojekt?



[Grechenig et. al.: Softwaretechnik. Pearson Studium, 2010]



Einführung – Top 10 Liste der Risikoelemente des Software Engineering

- Personalmängel
- Unrealistische Zeit- und Kostenpläne
- Entwicklung der falschen Funktionalität
- Unpassende Benutzerschnittstellen
- „Vergolden“ eines Systems
- Häufige Änderungen der Anforderungen
- Qualitätsmängel bei extern vergebenen Komponenten
- Qualitätsmängel bei extern vergebenen Aufgaben
- Unzureichende Produktionsleistungen
- Projektziele an der Grenze des „State-of-the-Art“

Nach: [Dumke, R.: Software Engineering. Vieweg Verlag, Wiesbaden, 2003]



Inhalt

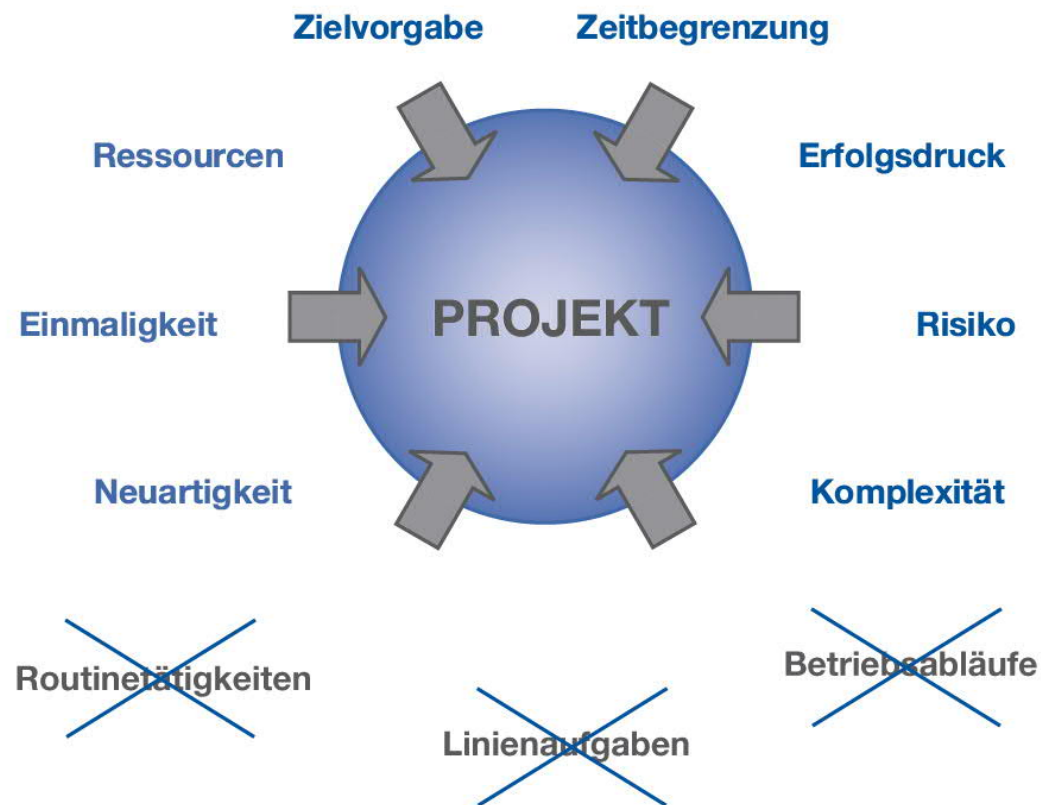
- Einführung
- **Das Projekt**
- Personen
- Prozesse
- Produkte
- Lernziele



Das Projekt - Definition

- In sich inhaltliches abgeschlossenes Vorhaben, in dem neuartige und unbekannte Probleme gelöst werden.
- Ein Projektteam.
- Klare Ziele.
- Beinhaltet eine oder mehrere Phasen.
- Unterschiedliche Methoden.
- Besonderes Risiko.
- Zeitlich begrenzt.
- Eigenes Budget.

Das Projekt



[Grechenig et. al.: Softwaretechnik. Pearson Studium, 2010]



Das Projekt – Merkmale eines Projekts

- Größe
- Dauer
- Anwendungsbereiche
- Quellen der Komplexität
- Zielsetzungen
- Ausgangs- & Zielprodukte
- Projekttypen

Merkmale eines Software-Projekts – Größe & Dauer & Anwendungsbereiche

Projektgrößen aus Softwaresicht (Fallbeispiele)

	<u>Jahre</u>	<u>#Team</u>	<u>PM</u>	<u>€ (2010)</u>
■ Sehr kleines Projekt	0,25	1	3	15.000
■ Kleineres Projekt	0,5	2	12	80.000
■ Mittleres Projekt	1,5	5	60	600.000
■ Große Hochschule	3,0	15	500	6 Mio.
■ Tickets Staatl. Bahn	3,5	40	1400	25 Mio.
■ ID Paraguay	2,5	120	3500	55 Mio.
■ eHealth-Germany	5,0	500	30000	600 Mio.

[Grechenig et. al.: Softwaretechnik. Pearson Studium, 2010]

Merkmale eines Software-Projekts – Größe & Dauer & Anwendungsbereiche

„Size creates Complexity“: Erläuterung und Grobschätzung
der relativen Aufwände abhängig von der Projektgröße

	PM	Aufw.	Dauer	Größensymptom (Bsp.)
■ Sehr kleines Projekt	3	0.5	0.8	User = AG = Develop.
■ Kleineres Projekt	12	1	1	Benutzerhandbuch
■ Mittleres Projekt	60	3	1.5	Reviews, autom. Tests
■ Große Hochschule	500	6	4	perm. User-Workshops
■ Tickets Staatl. Bahn	1400	15	4	Call-Center, 99,995%
■ ID Paraguay	3500	10	8	präventive Medienarbeit
■ eHealth-Germany	30000	50	20	Projekt wird zur Behörde

Auswirkungen von
Projektgröße und
-typ auf Dauer und
Aufwand für die
Umsetzung von
Anforderungen

[Grechenig et. al.: Softwaretechnik. Pearson Studium, 2010]



Merkmale eines Software-Projekts – Quellen der Komplexität 1(2)

- Jedes der bisher genannten Projektmerkmale kann als Quelle von Komplexität auftreten.
 - Projektgröße
 - falls die Größe für das Unternehmen ungewohnt ist.
 - Zielsetzungen
 - falls diese unklar sind, oder sich während der Projektlaufzeit ändern.
 - Domäne
 - falls die Begriffe und Sprache der Domäne nicht ausreichend geklärt sind.
 - Neue Technologien
 - erfordern ggf. neue Vorgehensweisen, zusätzliche Mitarbeiterschulungen usw.

Merkmale eines Software-Projekts – Quellen der Komplexität 2(2)

Projektkomplexität aus User- und Interessens-Dynamik Beispiele und Größenordnungen zur Veranschaulichung					
	Promotion	Core	User	Visitors	User-Kompl.
■ Sehr kleines Projekt	1	2	10	200	0.01
■ Kleineres Projekt	1	10	30	3000	0.25
■ Mittleres Projekt	4	100	2000	20.000	1
■ Große Hochschule	100	4000	40.000	1 Mio.	100
■ Tickets Staatl. Bahn	15	8000	4 Mio.	20 Mio.	200
■ ID Paraguay	50	15K	8 Mio.	> 8 Mio.	400
■ eHealth-Germany	1000	200K	80 Mio.	> 80 Mio.	4000

Abbildung 2.5: Projektkomplexität. Promotoren sind die Führungsspitzen, die hinter dem System stehen, Core-User die professionellen Anwender und Informationsbereitsteller, User bezeichnen die normalen Nutzer, Visitor besuchen das System gelegentlich, surfend oder zufällig. Der Wert für User-Komplexität ist ein aggregierter Expertenschätzwert aus allen vier Teilkomplexitäten. Als Faktorbasis fungiert das mittlere Projekt.

[Grechenig et. al.: Softwaretechnik. Pearson Studium, 2010]



Merkmale eines Software-Projekts – Zielsetzungen

- Beweggrund, warum ein Projekt durchgeführt werden soll:
 - ☐ Kurzfristige ökonomische Interessen,
 - ☐ Strategisches Investitionsprojekt,
 - ☐ Projekt in der Krise,
 - ☐ Forschungsprojekt,
 - ☐ ...



Merkmale eines Software-Projekts – Ausgangs- und Zielprodukte

■ Ausgangsprodukte

- ☐ Keine Vorgaben
- ☐ Produkte aus einem aktuellen Vorgängerprojekt
- ☐ Bestehendes System als Vorlage

■ Projekttypen


- ☐ Analyseprojekt
- ☐ Realisierungsprojekt
- ☐ Wartungsprojekt
- ☐ Gesamtprojekt
(beinhaltet Aspekte aller Projektarten)



Merkmale eines Software-Projekts – Projekttypen (nach Auftraggeber) 1(2)

Projekttyp	Ergebnis	Auftraggeber			
		internes Marketing	externer Kunde	internes Management	Verkauf / externer Kunde
Entwicklungsprojekt	Produkte, Systeme für den Markt	X	(X)	(X)	
Auftragsprojekt	Kundenspezifisches Softwaresystem		X		
EDV-Projekt	Datenverwaltung, Informationssysteme		(X)	X	
Systemprojekt	Industrieanlagen, technische Systeme			(X)	X

[Ludewig, Lichter: Software Engineering. dpunkt.verlag, 2010]



Merkmale eines Software-Projekts – Projekttypen (nach Auftraggeber) 2(2)

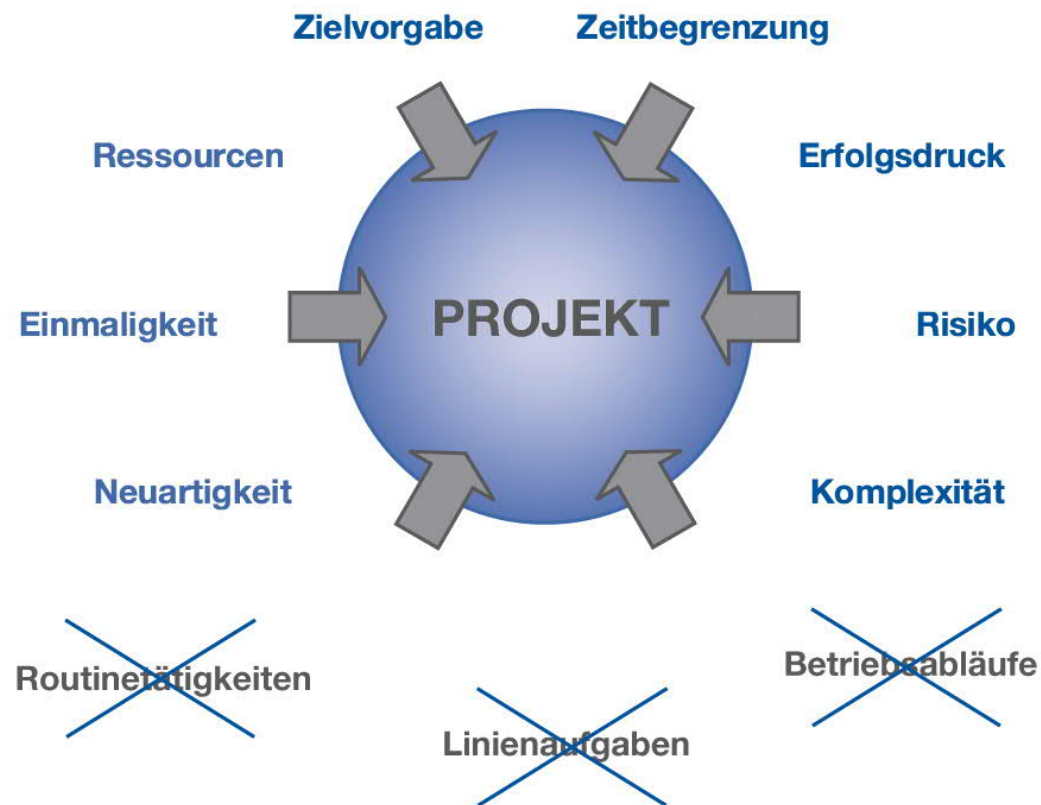
- Entwicklungsprojekt
 - „Standardsoftware“, „Shrinkware“
 - Auftraggeber: Marketingabteilung (Geld: eigenes Entwicklungsbudget)
- Auftragsprojekt
 - „Software nach den Wünschen des Auftraggebers“
 - Entwicklung auf der Basis eines Vertrags mit Lieferbedingungen, Zahlungsvereinbarungen, Pflichtenheft häufig Teil des Vertrages
- EDV-Projekt
 - „Software für den eigenen Bedarf“ (mögl. Ziele: Effizienzsteigerung, Rationalisierung)
 - Abrechnung zwischen den Abteilungen der Firma
- Systemprojekt
 - „komplexes System mit Softwareanteilen“ (z.B. Industrieanlage, Flugzeug)



Das Projekt – Merkmale eines Projekts

- Größe
- Dauer
- Anwendungsbereiche
- Quellen der Komplexität
- Zielsetzungen
- Ausgangs- & Zielprodukte
- Projekttypen

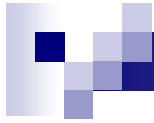
Das Projekt



[Grechenig et. al.: Softwaretechnik. Pearson Studium, 2010]

Faktoren eines Softwareprojekts





Inhalt

- Einführung
- Das Projekt
- **Personen**
- Prozesse
- Produkte
- Lernziele



Die Personen – Industrial Needs ...

... for software engineers coming freshly from university.

- Knowledge of state-of-the-art technology.
- Knowledge of theoretical background.
- Ability to understand new methods and tools.
- Familiarity with current major paradigms of SE.

- Understanding of the whole process of SE.
- Ability and modesty to do any job within that process.
- Ability and modesty to do any job within the team.
- Understanding and respecting the value of actual project experience.



Die Personen – Rollen je nach Projektgröße

Projekt-Größe	Rollen
Sehr klein	Keine besondere Rolleneinteilung notwendig.
Klein	Kunde; Management, Programmierer und Tester vereinigt in einer Person.
Mittel	Kunde, Projektleiter, Analytiker, Integrator, Programmierer und Tester.
Groß	Kunden, oberes Management, wirtschaftlicher Projektleiter, technischer Projektleiter, Gruppenleiter, Analytiker, Integrator, Programmierer, Tester, Dokumentierer, Qualitätssicherer.
Sehr groß	Alle Rollen aus Projektgröße "groß" und zusätzliche Rollen je nach Projekt (z.B. Berater, Domänenexperten, Technologieexperten usw.).



Die Personen – Rollen

■ Umfeld

- ☐ Kunde
- ☐ Management

■ Team

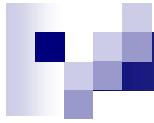
- ☐ Projektleiter
- ☐ Gruppenleiter
- ☐ Analytiker
- ☐ Integrator
- ☐ Programmierer
- ☐ Tester
- ☐ Qualitätssicherer



Die Personen – Rollen – Umfeld

- Kunde (ggf. auch mehrere Personen)
 - Kunde legt die Anforderungen festlegt.
 - Es sollte immer ein Hauptverantwortlicher und ein Vertreter benannt werden, die befugt sind Entscheidungen wirtschaftlicher und organisatorischer Natur treffen können.
 - Bei großen Projekten empfiehlt sich auch einen Vertreter der oder einen Kontakt in die Benutzergruppe zu bestimmen, um Anforderungen diskutieren/festlegen zu können.

- Management
 - Häufig Geschäftsleitung
 - Verantwortlich für die Zuteilung von Ressourcen (u.a. Mitarbeiter, Räume, Geräte, Zeit, ...).
 - Entscheidet bei Problemen auch über Fortführung oder Abbruch eines Projekts.



Die Personen – Rollen – Team (1)

■ Projektleiter

- ☐ ... hat wirtschaftliche und technische Verantwortung bei der Projektdurchführung.
- ☐ ... trifft autonome Entscheidungen im Rahmen des vorgegebenen Zeit- und Kostenplans.
- ☐ ... erstattet Kunde & Management Bericht bzgl. dem Projektfortschritt, Problemen, usw.

■ Gruppenleiter

- ☐ ... ist die Spitze der Entwickler seiner Gruppe.
- ☐ ... ist die erste Anlaufstelle bei technischen Problemen.
- ☐ ... teilt die Entwicklungsaufträge den Entwicklern der Gruppe zu.



Die Personen – Rollen – Team (2)

■ Analytiker

- ☐ ... erstellt das Analysemodell (als Teil der Anforderungen).
- ☐ ... besucht üblicherweise häufig den Kunden (zur Ermittlung der Anforderungen).
- ☐ ... ist im Idealfall jemand mit gutem Abstraktionsvermögen, d.h. kein typischer Entwickler, der sofort die Umsetzung im Blick hat.

■ Integrator

- ☐ ... befasst sich mit den technischen Anforderungen des Systems.
- ☐ ... untersucht die Schnittstellen (parallel zur Analysephase).
- ☐ ... ist zusammen mit dem Gruppenleiter hauptverantwortlich für den Entwurf.
- ☐ ... ist verantwortlich für Inbetriebnahme und Datenmigration beim Kunden.
- ☐ ... ist der technische Ansprechpartner für den Kunden.



Die Personen – Rollen – Team (3)

■ Programmierer

- ... erstellt hochwertigen Programmcode (inkl. Dokumentation, Tests, usw.)

■ Tester

- ... erstellt Testdaten und Testpläne.
- ... sollte eher wie ein Anwender denken und weniger wie ein Programmierer.
- ... führt Tests durch und schreibt Test- und Fehlerberichte.

■ Qualitätssicherer

- ... prüft die Einhaltung aller Vorgaben des Software-Entwicklungsprozesses.
- ... berichtet dem Projektleiter.
- ... erarbeitet Vorschläge für die Verbesserung des Entwicklungsprozesses.



Die Personen – Anzahl der Mitarbeiter nach Rollen

Unternehmensgröße	Projekt-leiter	Arbeits-gruppen-leiter	Entwick-ler	Spezia-listen	Qualitäts-sicherer
Groß (ab 120 Mitarbeiter)	8-10	15-20	50-70	5-12	4-10
Mittel (bis 40 Mitarbeiter)	3-5	5-6	15-20	2-5	2-5
Klein (bis 15 Mitarbeiter)	1-2	2-3	5-7	0-2	0-1
Sehr klein (bis 5 Mitarbeiter)	1	1	5	0	0
Ungefähre Relation im Vergleich zu Projektleitern	1	2-3	5-10	0,5-1	0,5-1



Die Personen – Arbeitsgruppe 1(8)

- Zur Bedeutung von Arbeitsgruppen für die einzelne Person:
 - *Stellen Sie sich vor, Ihr Chef hat Ihnen gerade eine Spezifikation auf den Tisch gelegt und gefragt: „Wie lange brauchen Sie mit einer weiteren Person, um das zu erledigen?“ Welche Frage stellen Sie als Erstes?*
 - *Würden Sie fragen: „Können wir objektorientierte Methoden verwenden?“ oder „Welches CASE-System können wir kaufen?“ Natürlich nicht. Ihre erste Frage lautet: „Wer ist die andere Person?“*

[T. DeMacro. Warum ist Software so teuer? Und andere Rätsel des Informationszeitalters. Hanser Verlag, München, 1997.]



Die Personen – Arbeitsgruppe 2(8)

- Arbeitsgruppen werden nach Bedarf zusammengestellt. Sie können in einer Zusammensetzung nur ein Projekt oder nacheinander für mehrere Projekte durchführen.
- (Theoretische) Modelle für die Zusammensetzung von Arbeitsgruppen:
 - Anarchische Arbeitsgruppe
 - Demokratische Arbeitsgruppe
 - Chefprogrammierer-Team
 - Typische Gruppenstruktur



Die Personen – Arbeitsgruppe 3(8)

- Anarchische Arbeitsgruppe
 - Entwickler arbeiten autonom nach eigenen Vorgaben
 - Vorteile:
 - Entwickler arbeiten selbstbestimmt, keine Hierarchie-Probleme, Kaum bürokratischen Hindernisse.
 - Nachteile:
 - Standards/Normen lassen sich nicht durchsetzen.
 - Launen der Mitarbeiter bestimmen Planung, Abstimmung zwischen den Mitarbeitern und die Einführung neuer Methoden und Technologien
 - Organisation ist nicht lernfähig.
 - Offiziell gibt es diese Arbeitsgruppe nicht!
 - Häufig in Organisationen mit schwacher Führungsstruktur anzutreffen!
 - Meist als Demokratische Arbeitsgruppen oder Chefprogrammierer-Teams getarnt!



Die Personen – Arbeitsgruppe 4(8)

■ Demokratische Arbeitsgruppe

□ Annahme:

- Programmierer hat eine Beziehung zu seinem Produkt und sucht daher nur ungern und ungenau nach Fehlern in seinem Code.

□ Idee:

- Die Beziehung aufzulösen, indem alle Gruppenmitglieder gleichberechtigt sind. Es gibt keinen festgelegten Gruppenleiter. Die Gruppe muss eine gemeinsame Vision besitzen und das Suchen und Finden von Fehlern darf nicht „persönlich“ genommen werden.

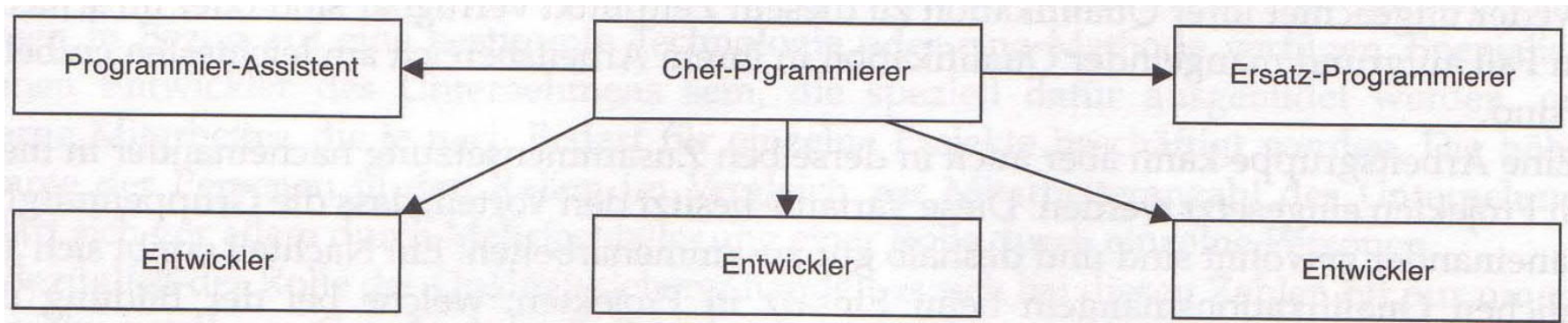
□ Probleme:

- Viele Kommunikationswege, da jeder mit jedem sprechen muss.
- Gruppe funktioniert nur, solange es keine Beförderungen oder Rangordnungen in der Gruppe gibt.

Die Personen – Arbeitsgruppe 5(8)

■ Chefprogrammierer-Team

- Vergleichbar mit Operationsteam, in dem ein Chefarzt von einem Team von Spezialisten unterstützt wird.



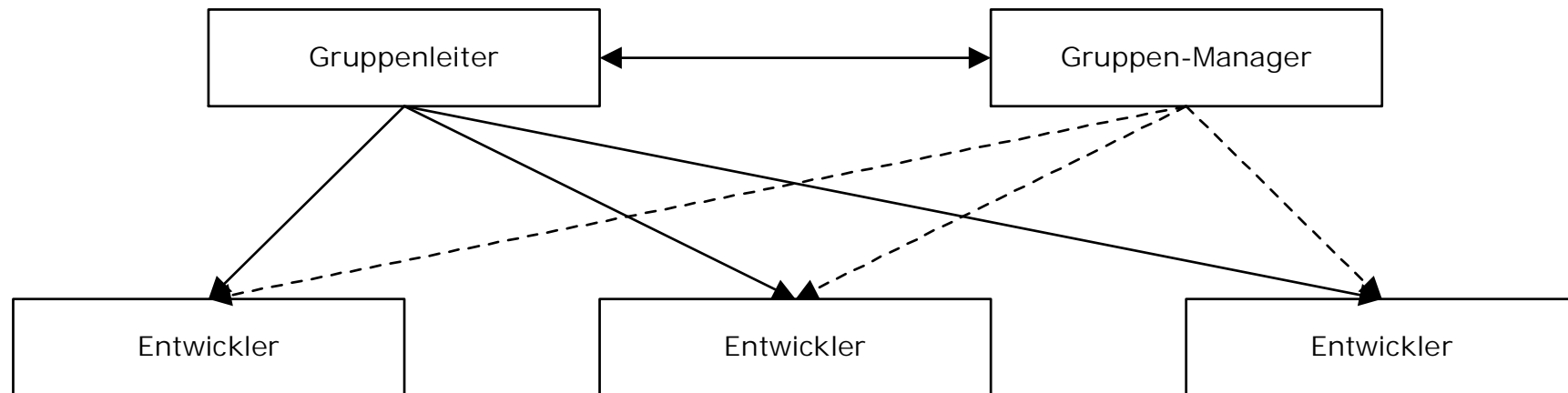


Die Personen – Arbeitsgruppe 6(8)

- Chefprogrammierer-Team (cont.)
 - Chef-Programmierer
 - Guter Manager und talentierter Programmierer.
 - Ersatz-Programmierer
 - Muss den Chef-Programmierer bei dessen Ausfall administrativ und technisch ersetzen können.
 - Programmier-Assistent
 - Erledigt die administrativen Aufgaben und kümmert sich um technische Belange wie Archivierung von Programmversionen und Dokumentation des Projekts.
 - Problem:
 - Wie findet man einen Chef-Programmierer (Top-Manager und -Programmierer in einer Person) und einen gleichwertigen Ersatz-Programmierer (der sich dann auch mit der Rolle des Ersatzprogrammierers begnügt).

Die Personen – Arbeitsgruppe 7(8)

■ Typische Gruppenstruktur



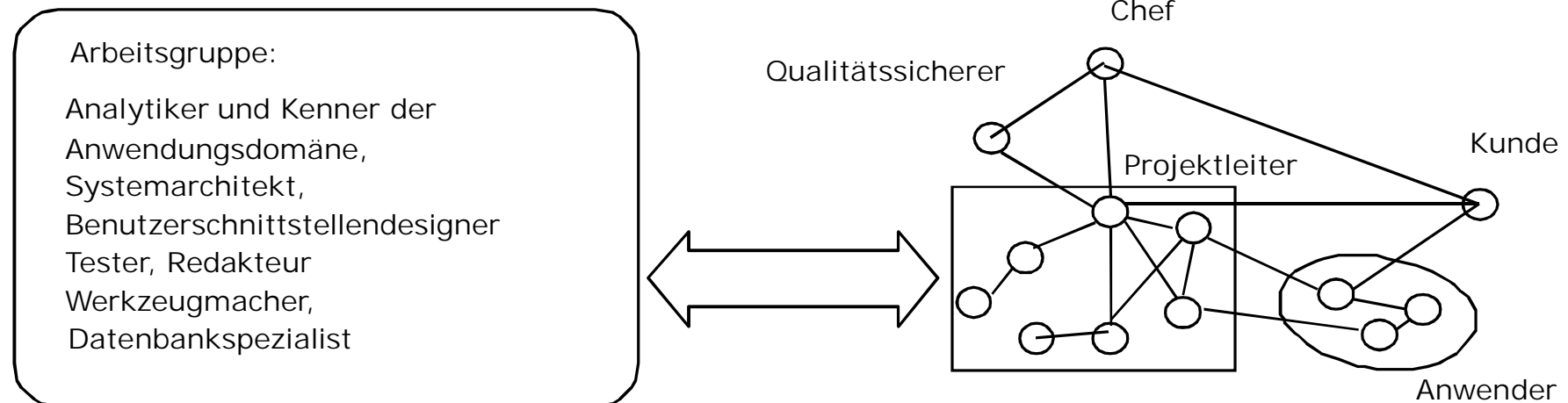


Die Personen – Arbeitsgruppe 8(8)

- Gruppenleiter
 - entscheidet über technische Belange.
- Gruppen-Manager
 - übernimmt die organisatorischen und wirtschaftlichen Aufgaben.
- Gruppenleiter und Gruppen-Manager
 - stimmen sich ab und teilen die Gruppe nach Bedarf ein.
- Entwickler
 - sind Spezialisten in verschiedenen Rollen (Analytiker, Programmierer, Tester usw.)

Die Personen – Arbeitsgruppe, Unternehmen & Kunde

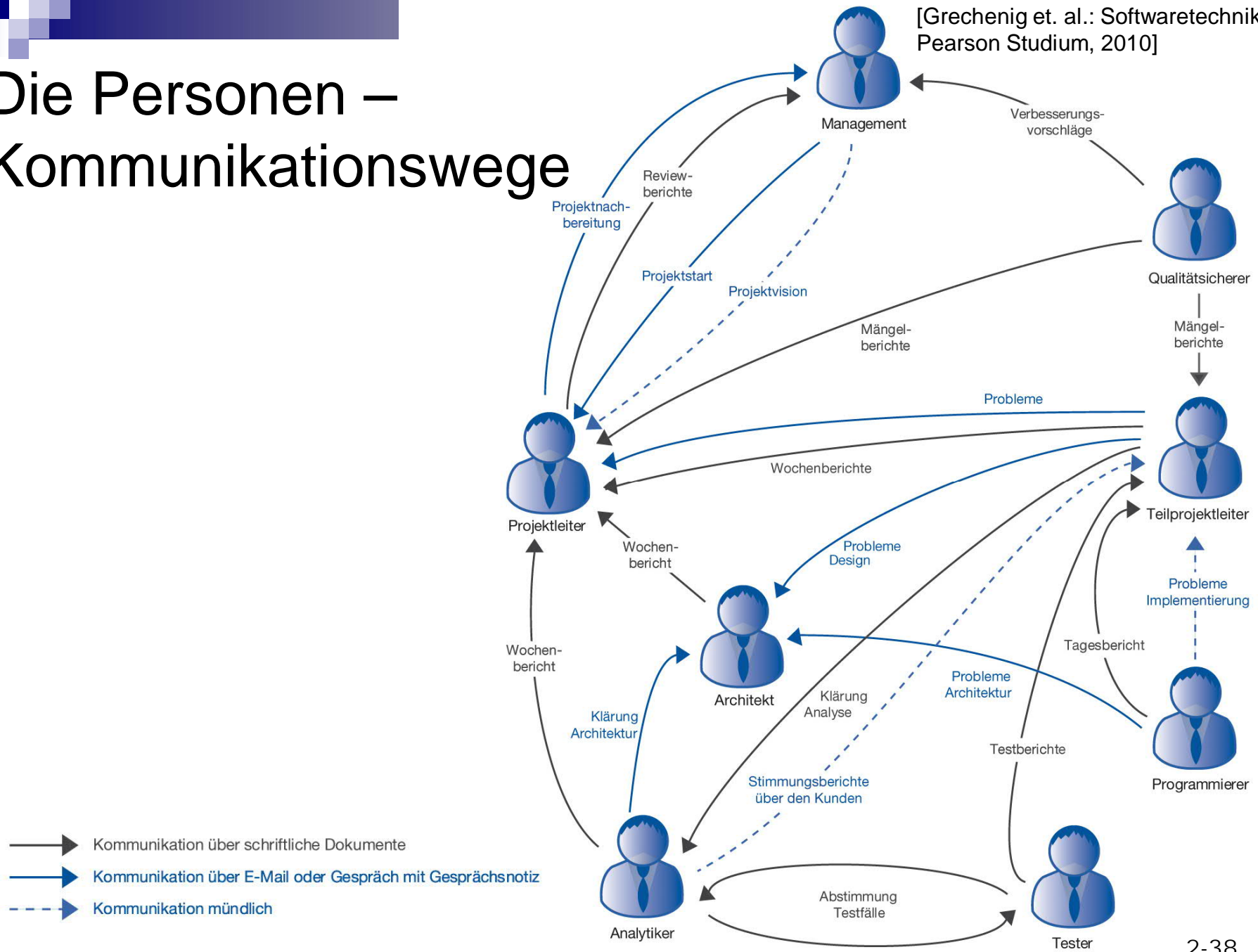
- Arbeitsgruppe und ihre Einbindung im Unternehmen und ihre Verbindung zum Kunden.






Die Personen – Kommunikationswege

[Grechenig et. al.: Softwaretechnik.
Pearson Studium, 2010]





Die Personen – 1(12)

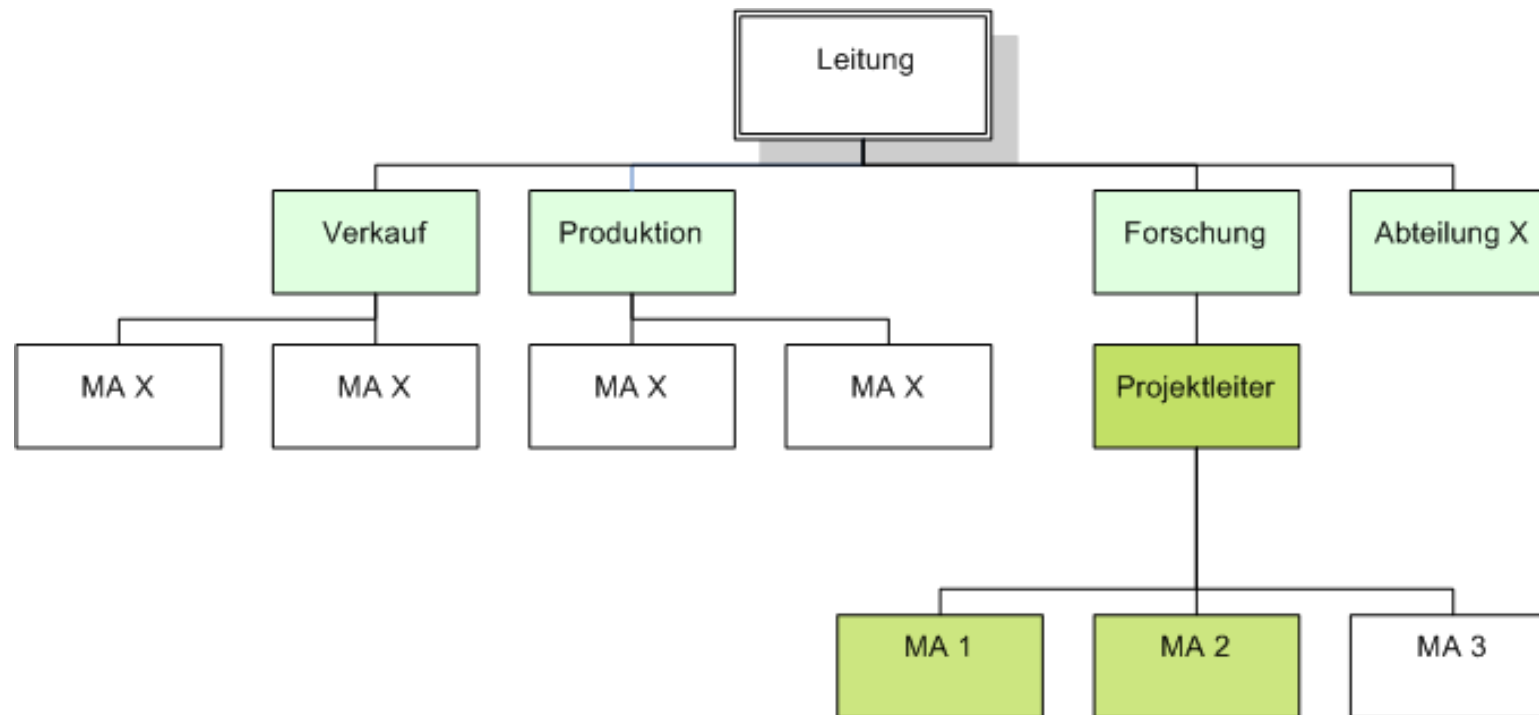
Projekt- & Primärorganisation


- Primärorganisation
 - Organisationsstruktur des Herstellers
 - Aufgabenverteilung/Beziehungen zwischen den Mitarbeitern
- Sekundärorganisation
 - Projektgruppe
- Mögliche Organisationsformen
 - Linienprojektorganisation
 - Stabslinienprojektorganisation (Einflussprojektorganisation)
 - Matrixprojektorganisation
 - Reine Projektorganisation
 - Auftragsprojektorganisation
 - Projektgesellschaft

Die Personen – 2(12)

Projekt- & Primärorganisation

■ Linienprojektorganisation





Die Personen – 3(12)

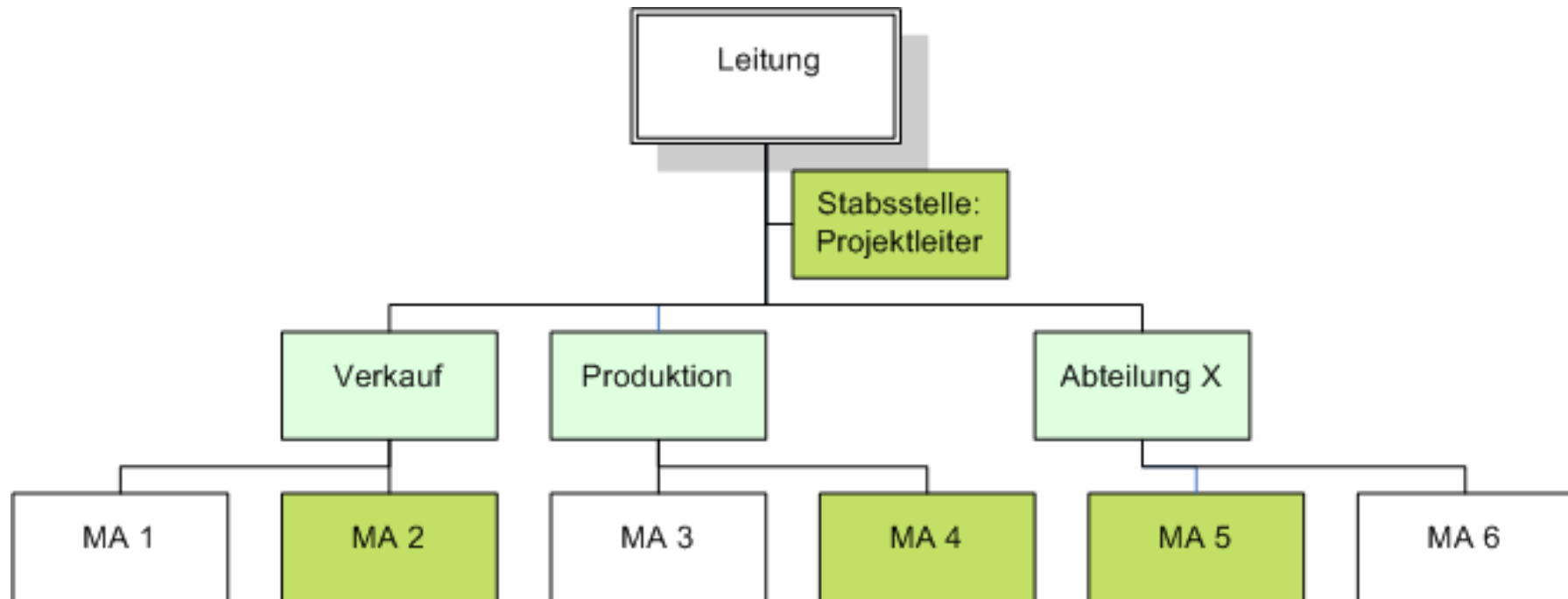
Projekt- & Primärorganisation


- Linienprojektorganisation (cont.)
 - Keine eigene Projektorganisation. Projekte werden in die vorgegebene Primärorganisation integriert.
 - Vorteile
 - Geringer Aufwand mit kurzen Kommunikationswegen.
 - Kein Wechsel von Kompetenzen.
 - Nachteile
 - Bedeutung eines Projektes wird nicht durch die Organisation sichtbar.
 - Weniger Akzeptanz für ein Projekt.
 - Konflikt zwischen Tages- und Projektarbeit.

Die Personen – 4(12)

Projekt- & Primärorganisation

- Stabslinien- oder Einflussprojektorganisation





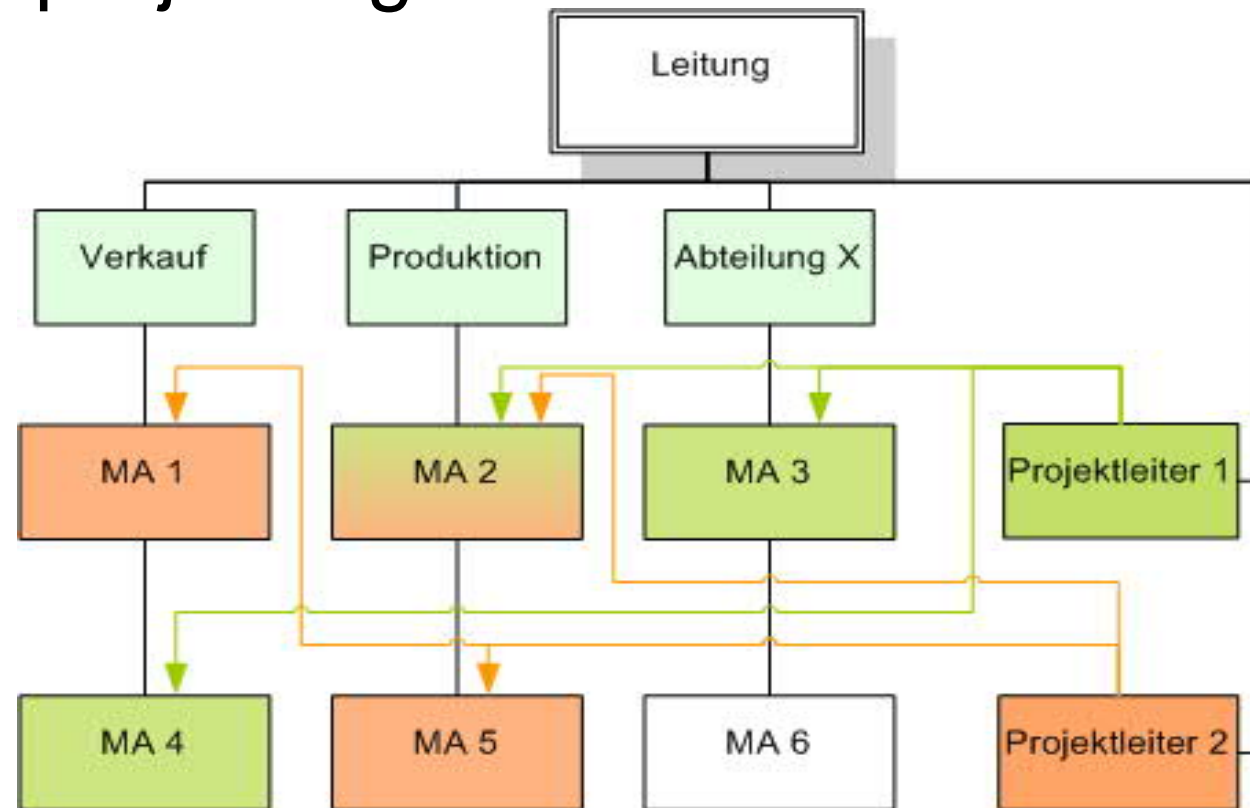
Die Personen – 5(12) Projekt- & Primärorganisation


- Stabslinien- oder Einflussprojektorganisation
 - Projektleitung erfolgt durch eine Stabsstelle
 - Stabsstelle:
 - Direkt der Unternehmensführung zugeordnet.
 - Keine formale Weisungsbefugnis gegenüber Linienmitarbeitern.
 - Vorteile
 - Bedeutung des Projekts wird durch Stabsstelle deutlich
 - Kein Eingriff auf Linienorganisation
 - Geringer Aufwand
 - Flexibler Personaleinsatz
 - Nachteile
 - Keine (formale) Weisungsbefugnis
 - Lange Kommunikationswege

Die Personen – 6(12)

Projekt- & Primärorganisation

■ Matrixprojektorganisation





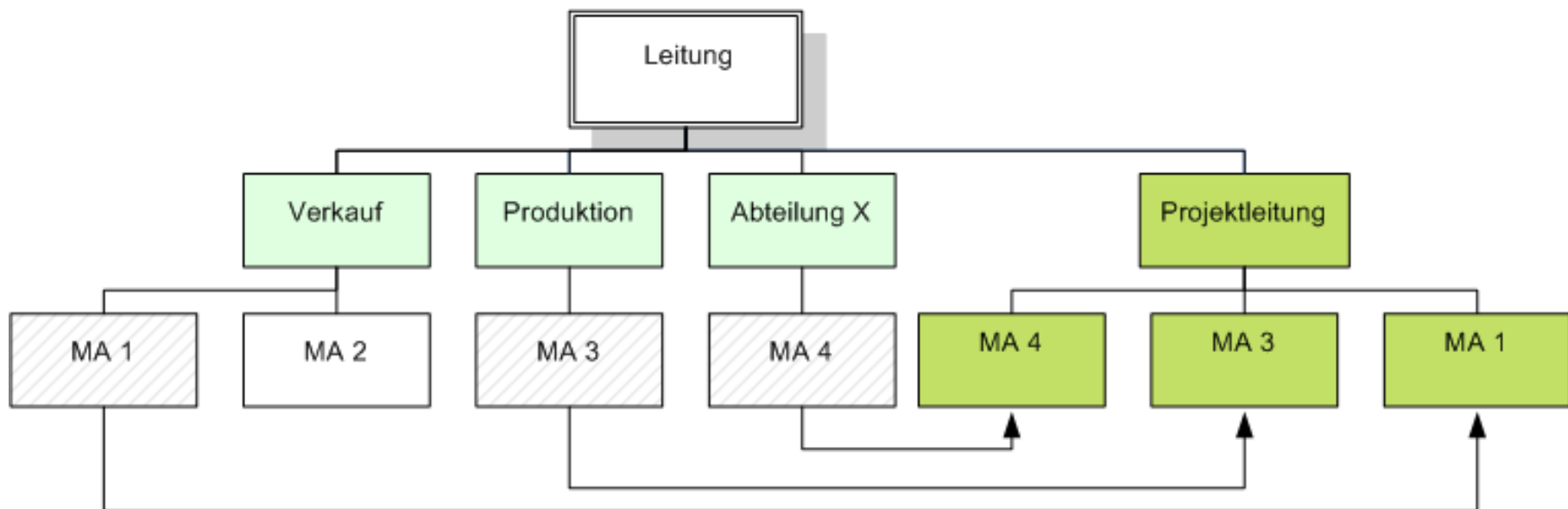
Die Personen – 7(12) Projekt- & Primärorganisation


- Matrixprojektorganisation (cont.)
 - Prinzip der Mehrfachunterstellung (Mehrliniensystem)
 - Projektleiter erhält fachliche Weisungsbefugnis (jedoch keine personellen oder disziplinarischen Kompetenzen)
 - Vorteile
 - Flexibler Personaleinsatz
 - Gezielter Einsatz von Spezialwissen
 - Nachteile
 - Mehrfachunterstellung (Probleme mit Weisungsbefugnis)
 - Lange und komplizierte Kommunikationswege

Die Personen – 8(12)

Projekt- & Primärorganisation

■ Reine Projektorganisation



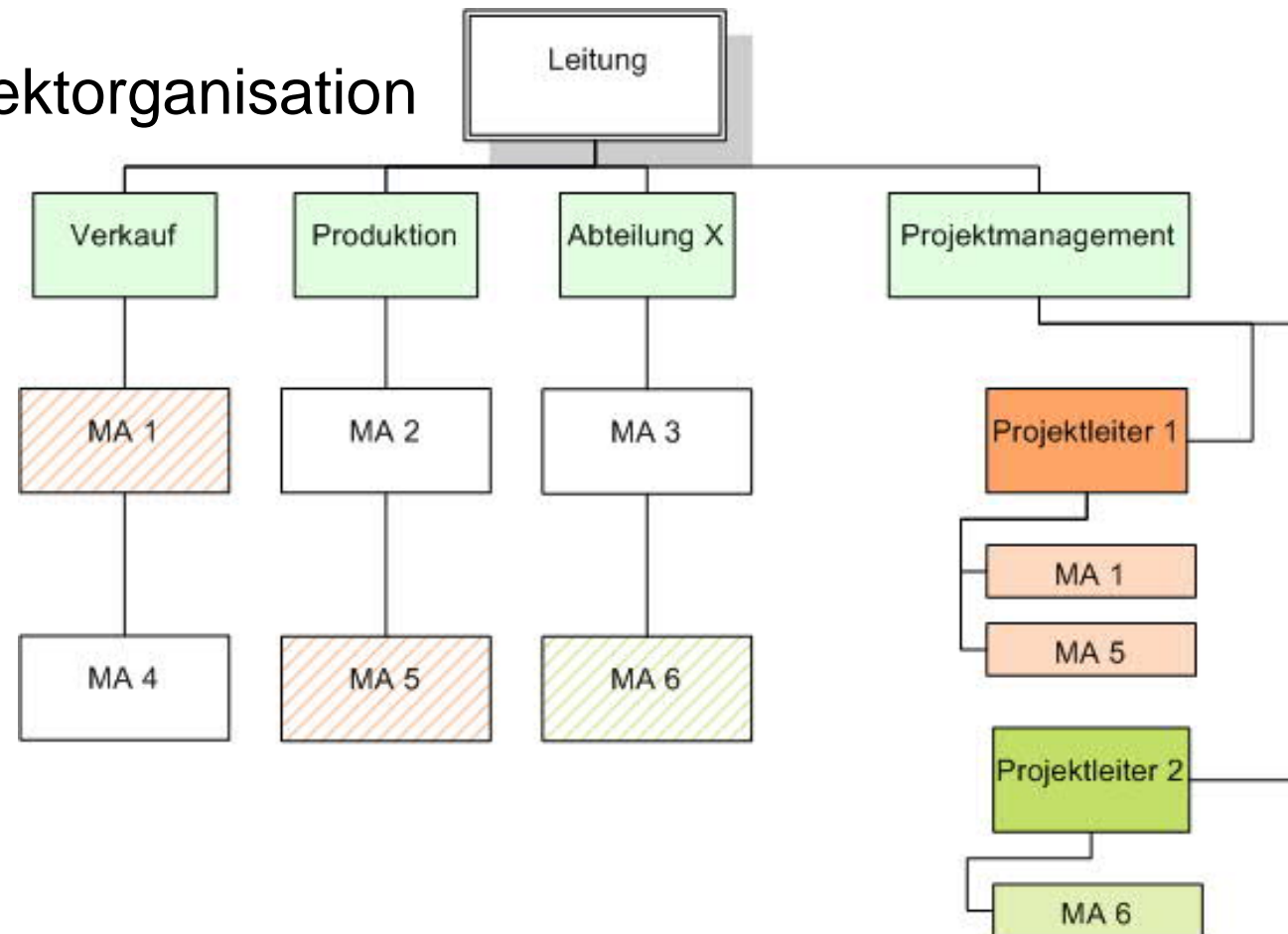



Die Personen – 9(12) Projekt- & Primärorganisation

- Reine Projektorganisation (cont.)
 - Projektteam wird als temporäre Struktur in die Firmenorganisation einbezogen. Projektleiter bekommt Fach- und Linienkompetenzen.
 - Vorteile
 - Feste Zuordnung & damit ständige Verfügbarkeit von Ressourcen
 - Kurze Kommunikationswege
 - Hohe Mitarbeitermotivation durch Zugehörigkeitsgefühl
 - Schnelle Entscheidungen, schnelles Auffinden von Fehlern
 - Nachteile
 - Große Umstrukturierungen notwendig
 - Festes Personal kann an anderer Stelle fehlen
 - Probleme bei Wiedereingliederung in ehemalige Abteilung

Die Personen – 10(12) Projekt- & Primärorganisation

- Auftragsprojektorganisation






Die Personen – 11(12)

Projekt- & Primärorganisation

- Auftragsprojektorganisation (cont.)
 - Projektleitung wird als „permanente“ Abteilung in die Primärorganisation integriert.
 - Projektleitungskompetenz der Firma wird „gebündelt“ und kann gezielt verbessert werden.
 - Ansonsten gleiche Vor- und Nachteile wie bei der reinen Projektorganisation



Die Personen – 12(12)

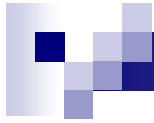
Projekt- & Primärorganisation

■ Projektgesellschaft

- ☐ Projektorganisation wird Primärorganisation
- ☐ Gründung einer Gesellschaft mit dem Ziel der Projektdurchführung
- ☐ Projektleiter = Unternehmensleiter

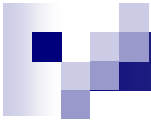
Faktoren eines Softwareprojekts





Inhalt

- Einführung
- Das Projekt
- Personen
- **Prozesse**
- Produkte



Vorgehens- & Prozessmodelle – Software-Life-Cycle 1(2)

- Wiederkehrende Schritte bei der Softwareentwicklung:

Analyse

Spezifikation der Anforderungen

Grobentwurf, Spezifikation der Module

Feinentwurf

Codierung und Modultest

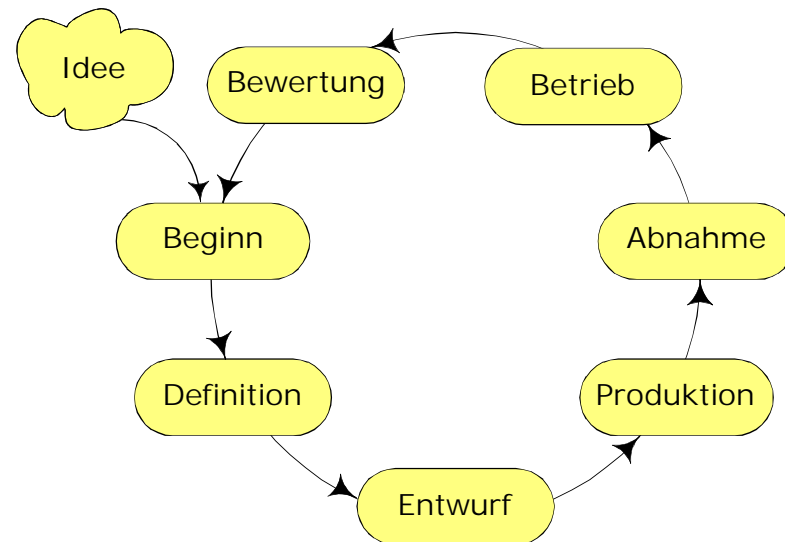
Integration, Test, Abnahme

Betrieb, Wartung

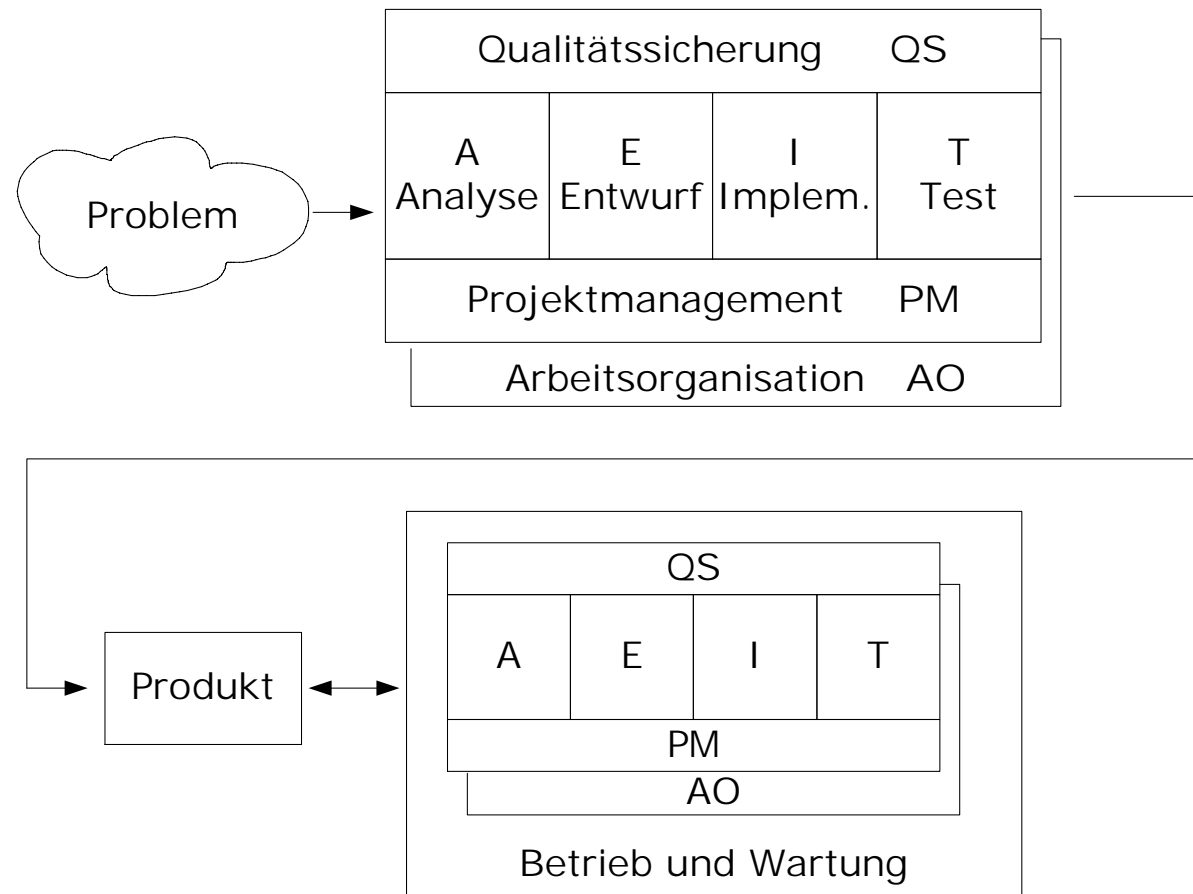
© Ludewig, Lichter, 2006

Vorgehens- & Prozessmodelle – Software-Life-Cycle 2(2)

- Def. IEEE Glossar
(IEEE Std. 610.12, 1990):
 - The period of time that begins when a software product is conceived and ends when the software is no longer available for use. The software life cycle typically includes a concept phase, requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and, sometimes, retirement phase.
 - Note: These phases may overlap or be performed iteratively.



Vorgehens- & Prozessmodelle – Basismodell

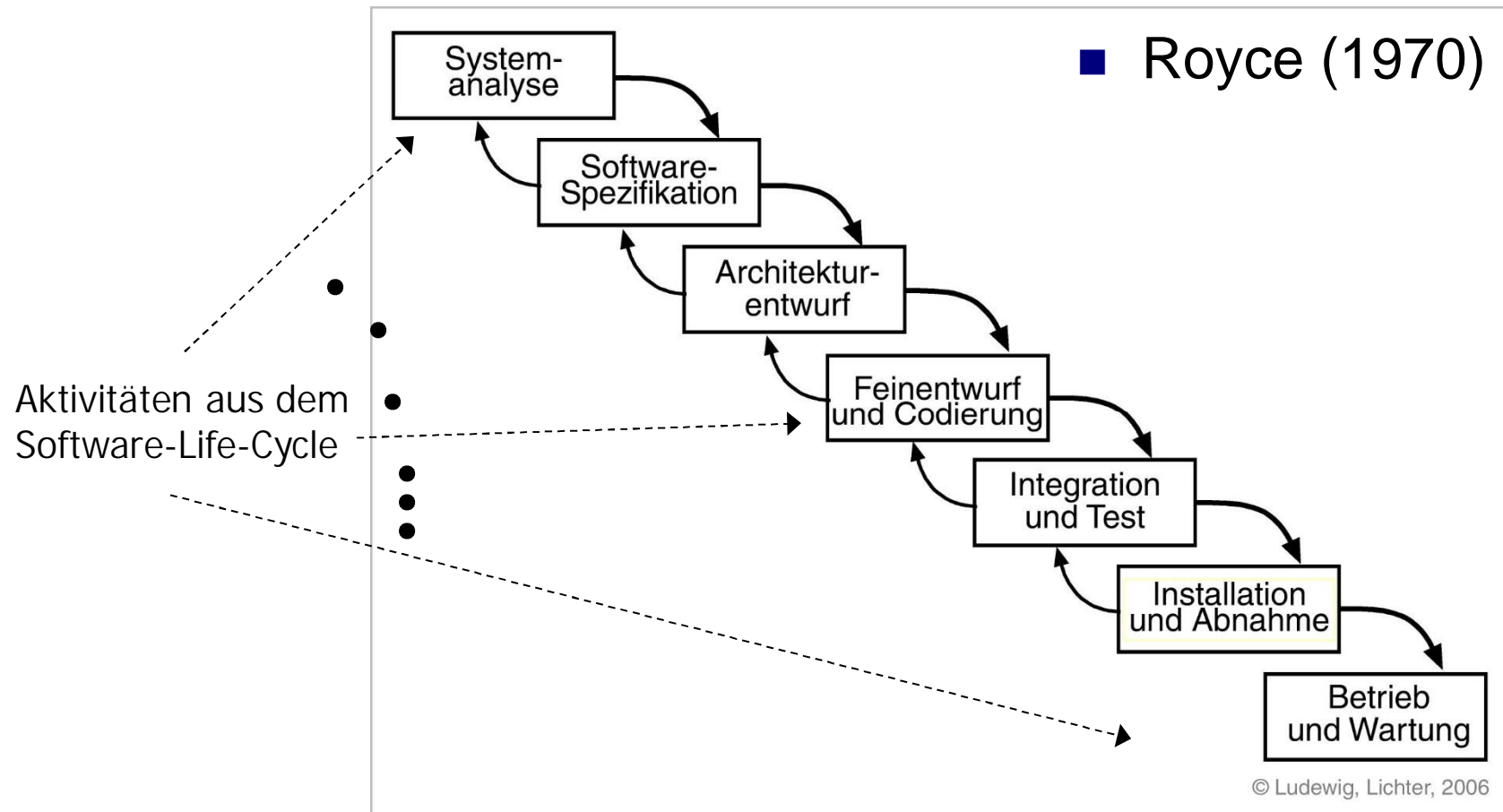




Vorgehens- & Prozessmodelle – Vorgehensmodelle

- **Strategien** für die Durchführung eines Projektes, z.B.
 - Build-and-Fix (Code-and Fix)
 - Wasserfallmodell
 - Prototyping
 - Nichtlineare Vorgehensmodelle
 - Rapid Prototyping
 - Evolutionäre Software-Entwicklung
 - Iterative Software-Entwicklung
 - Inkrementelle Software-Entwicklung
 - Treppenmodell
 - Phasenmodell

Beispiel: Vorgehensmodell – Wasserfallmodell

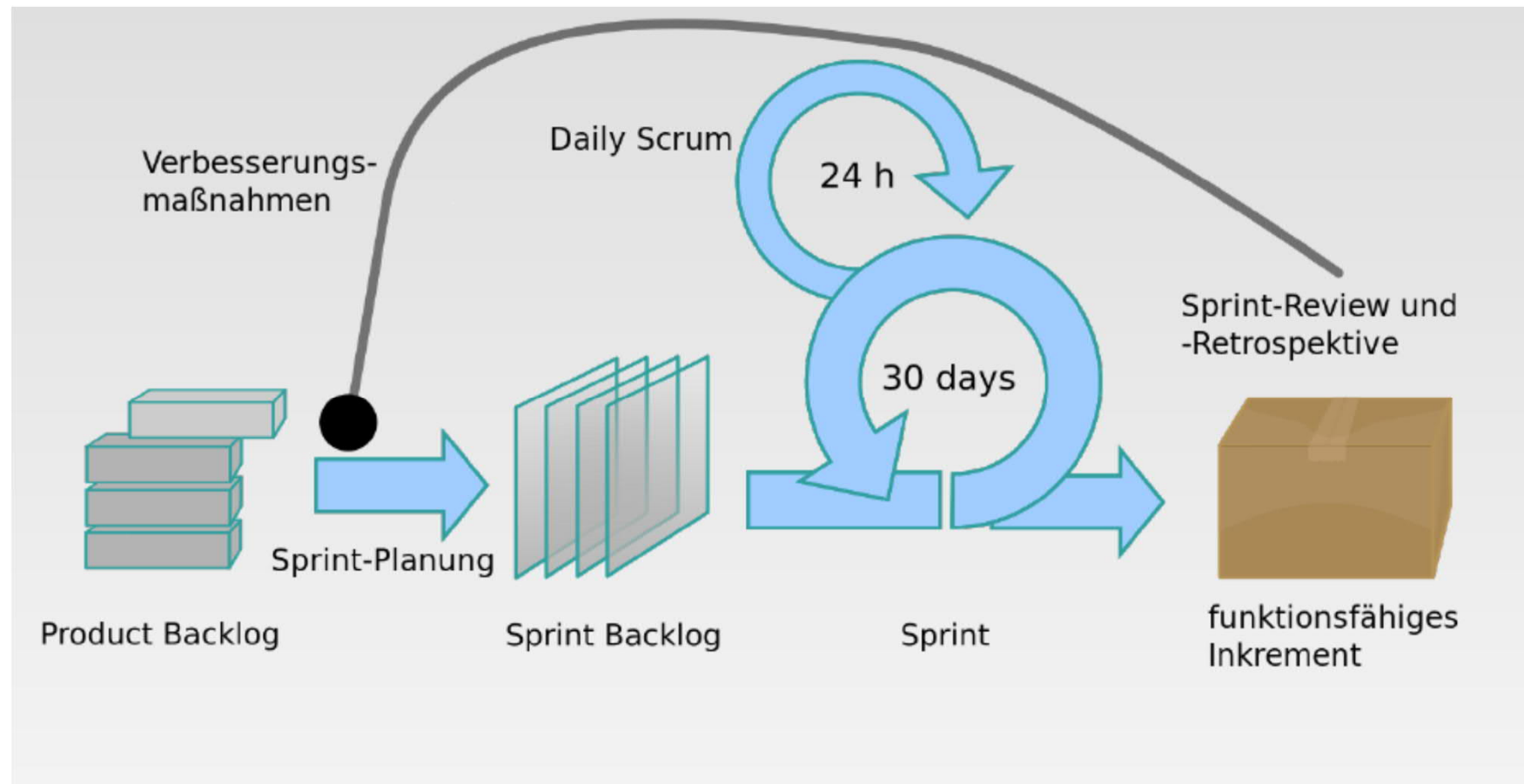




Vorgehens- & Prozessmodelle – Prozessmodelle

- **detaillierte Anweisungen** zur Durchführung von Softwareentwicklungsprojekten (Aussagen zu Organisation, Verantwortlichkeiten, zu erstellenden Dokumenten, zu verwendenden Methoden, usw.), z.B.
 - V-Modell
 - V-Modell XT
 - Unified Process
 - Agile Modelle
 - eXtreme Programming (XP)
 - Scrum

Beispiel Prozessmodell – Scrum





Erwartungen an Vorgehens- und Prozessmodelle

- Gemeinsame Sprache für Projekte.
- Weniger Nacharbeit.
- Vereinfachung von Routineabläufen.
- Basis kontinuierlicher Verbesserung.
- Wettbewerbsvorteil Zertifizierung.
- Hilfe für wenig erfahrene Projektmanager.
- Rascher Nutzen (z.B. Schulung neuer Mitarbeiter).
- Rücksicht auf typische Sonderfälle der Entwicklerrealität.

Faktoren eines Softwareprojekts

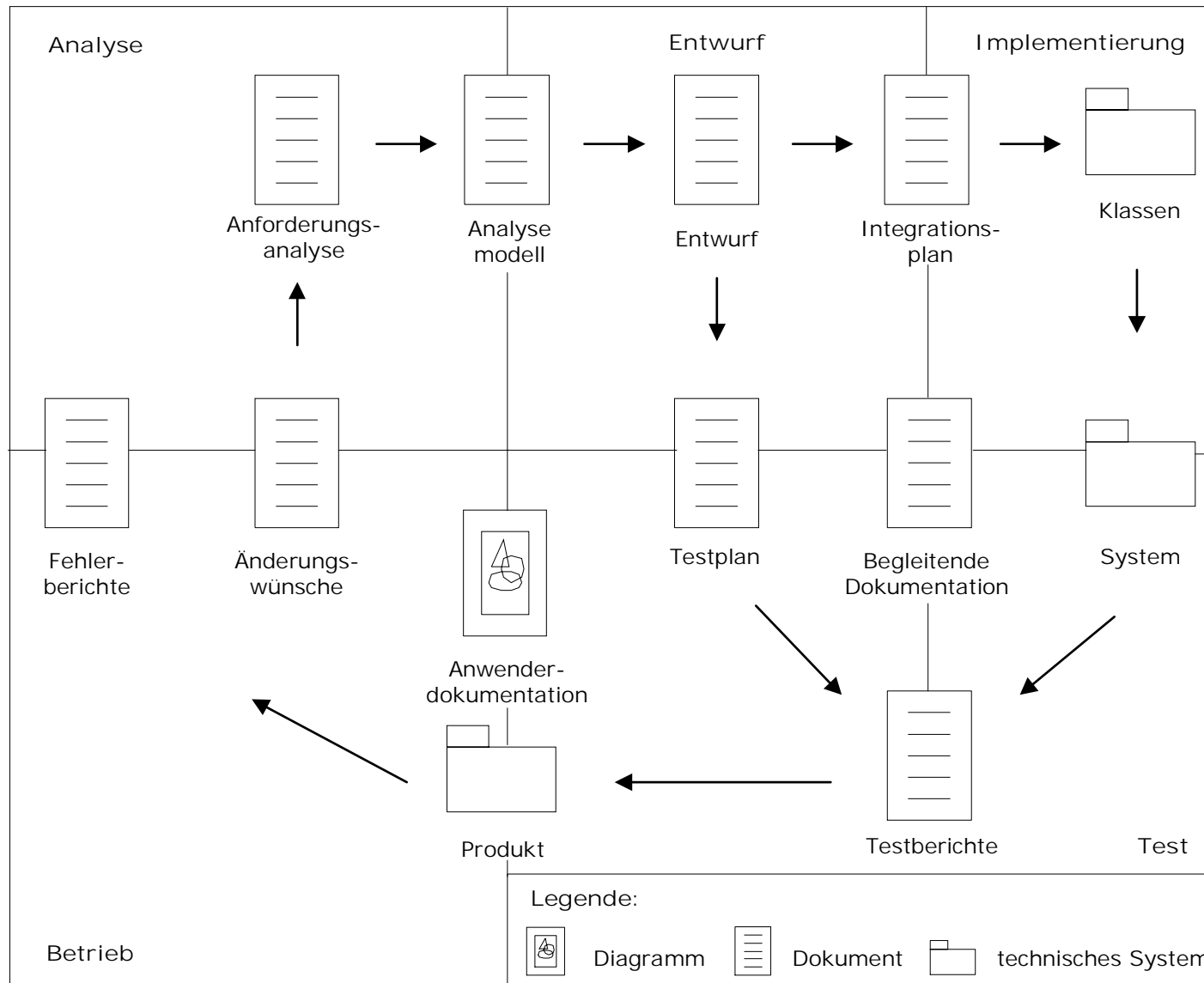




Inhalt

- Einführung
- Das Projekt
- Personen
- Prozesse
- **Produkte**
- Lernziele

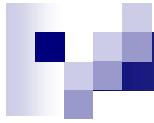
Produkte – Zwischen- und Endprodukte





Produkte

- Produkte stellen die überprüfbaren Resultate eines Projektes dar.
- Produktmerkmale
 - ☐ Zweck
 - ☐ Zielpublikum
 - ☐ Art (Doku, Techn. ...)
 - ☐ Detailgrad
 - ☐ Reifestufe
 - ☐ Abnahmekriterien
 - ☐ Referenzen
 - ☐ Erstellungszeitpunkt



Produkte – Formaler Rahmen

- Begriffe
- Dokumentationsrichtlinien
- Dokumentationsstruktur
- Produktvorlagen
- Checklisten
- Versionsmanagement
- Änderungsmanagement
- Archivierungsmanagement



Produkte – Qualitätsmerkmale

■ Dokumente

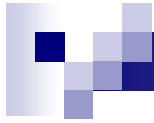
- ☐ Vollständigkeit
- ☐ Korrektheit
- ☐ Konsistenz
- ☐ Verständlichkeit
- ☐ Änderbarkeit

■ Technische Produkte

- ☐ Benutzerfreundlichkeit
- ☐ Korrektheit
- ☐ Zuverlässigkeit
- ☐ Effizienz
- ☐ Wartungsfreundlichkeit

Faktoren eines Softwareprojekts





Inhalt

- Einführung
- Das Projekt
- Personen
- Prozesse
- Produkte
- **Lernziele**



Lernziele

■ Softwareprojekte

- ☐ Was ist ein Projekt (Definition)?
- ☐ Was ist das Besondere an Softwareprojekten?
- ☐ Was sind die Merkmale von Softwareprojekten?

■ Personen

- ☐ Welche Rollen gibt es in Softwareprojekten?
- ☐ Wie setzen sich Arbeitsgruppen zusammen?
- ☐ Wie lassen sich Arbeitsgruppen in Unternehmen einbinden?



Lernziele (2)

■ Vorgehens- und Prozessmodelle

- ☐ Was ist der Software-Life-Cycle?
- ☐ Wie sieht das Basismodell für die Softwareentwicklung aus?
- ☐ Welche Erwartungen hat man an Vorgehens- und Prozessmodelle?

■ Produkte

- ☐ Welche Merkmale haben Produkte in der Softwareentwicklung?
- ☐ Welchen formalen Rahmen benötigt man für die Produkterstellung?
- ☐ Welche Qualitätsmerkmale sollten Produkte erfüllen?