

Software Auftragswesen

„Zur Abrechnung von IT-Dienstleistungen“

Eine Ringvorlesung für die TU Dresden

am 23.10.2017

von Harry M. Sneed

Reverse-Engineering Toolentwickler für VW-Finanz

Software Tester für ATOS Österreich

Migrationsberater für das Land Burgenland

Lehrbeauftragter für Software Engineering an den
Universitäten Szeged, Bialystok & Dresden sowie an
den Fachhochschulen Wien & Hagenberg

GIG–Das Wirtschaftsmodell der Zukunft

- Was ist GIG?
- Wenn Du es nicht weisst, solltest Du es schnell lernen, denn GIG wird Deine Zukunft sein!!
- Schaue in Google nach.

Auslaufmodell Festanstellung

- **Wiener Kurier Zeitung von 21. Oktober 2017: Bezahlt wird nicht mehr nach Stunden sondern nur noch nach Auftrag (GIG).**
- **Den Zuschlag erhält der bestbewertete Anbieter, statt dem Arbeitsrecht gelten die allgemeinen Geschäftsbedingungen der Plattform.**
- **Die sogenannte Gig-Economy ist auf dem Vormarsch. Internet-Plattformen wie Upwork, Clickwork, MyHammer, oder Amazons Mechanical Turk vermitteln an eine große Anzahl von Personen („the crowd“) Tätigkeiten, die bisher normale Arbeitsverhältnisse waren.**
- **Die Palette reicht von Übersetzungen, Graphikdesign, Bauplanung, Schreibaarbeiten, Buchhaltung bis zur Programmierung und Programmtest.**

Wo bleibt der Arbeitnehmer

- **Die Digitalisierung beschleunigt nicht nur die Verschiebung weg vom Angestelltenverhältnis hin zur atypischen Beschäftigung, auch der klassische, sozialrechtlich abgesicherte Arbeitnehmerbegriff gerät ins Wanken.**
- **18% der Österreicher haben laut Umfrage der Arbeitskammer schon für eine Crowd working Plattform gearbeitet, die Meisten davon in Nebenerwerb. Davon leben könnten die Wenigsten, bestätigt Ursula Huws von der britischen Universität Hertfordshire. Sie hat Crowd-Working in 7 EU Ländern untersucht. Ihr Befund: Die Praktiken der Online-Plattformen würden sich zunehmend auch auf andere Wirtschaftssektoren ausbreiten und dadurch das moderne Arbeitsleben bestimmen.**
- **„Wer ist in Zukunft eigentlich noch Arbeitnehmer?“ fragte WiFo Chef Badelt und fügt hinzu „Wenn jemand von einem Mini-Einkommen zum nächsten hechelt, geht sich das für die Pension nicht aus. Badelt sieht in den neuen Arbeitsformen eine der größten Herausforderungen für den Sozialstaat überhaupt.**
- **Die aus noch aus dem Industrie-Zeitalter stammenden Begriffe „Arbeitszeit“ und „Arbeitsort“ müssen an die digitale Welt angepasst werden.**

Requiem für die Angestellten

- Die Tage der Angestellten sind gezählt. Angestellt werden künftig nur noch die Manager
- Werktätigen wie Designer, Entwickler und Tester werden gezwungen selbstständig zu werden.
- Die in dem Kurier beschriebene GIG Arbeitsverhältnisse sind die Folge wirtschaftlicher Zwänge, die schon lange in der IT-Welt vorhanden sind.
- Das geht aus den folgenden Sprüchen hervor.

Wichtige Sprüche zu diesem Thema

- You get what you pay for! If you pay for Time, you get Time, If you pay for a result, you get some result!
- Only the results count, not the way you produce them
- Software Developers are paid to deliver executable Code, not dead Documents.
- Requirement Specifications are never finished! Therefore they must be made adaptable.
- Models are only useful if they correspond to the real World, i.e Code and Data.
- The Test is the ultimate barrier to success It must be well planned from the very Beginning – Test Driven Development.

Software Dienstleistungen

- Bisher wurde ausschließlich nach Stunden abgerechnet
- Mitarbeiter und Auftragnehmer sind angehalten Stundenberichte zu schreiben. Meistens sind sie gefälscht.
- Es gebe keinen Zusammenhang zwischen den gemeldeten Stunden und der geleisteten Arbeit.
- Wir brauchen einen neuen Modus um geleistete Arbeit abzurechnen.
- Nur die Ergebnisse zählen, es müsste nach Ergebnis und nicht nach Zeit bezahlt werden.

Was hat sich geändert ?

- Die Projekte sind mehr verteilt
- Immer mehr Arbeiten werden nach Außen vergeben
- Mitarbeiter dürfen im Home Office arbeiten.
- Der Unterschied in der Leistung der beteiligten Mitarbeiter wird immer größer.
- Die marxistische These dass ein Produkt so viel Wert ist wie die darin investierte Arbeit wurde widerlegt.
- Es gibt immer weniger Zusammenhang zwischen der geleisteten Arbeitsstunden und dem Wert einer Dienstleistung.
- Der Wert einer Dienstleistung wird allein durch die Nachfrage bestimmt.

Auftragsvereinbarungen

- Software Dienstleistungen müssten beauftragt werden.
- Vor der Erteilung eines Auftrages ist der Wert einer Dienstleistung auszuhandeln.
- Es stellt sich die Frage – was ist der Nutzwert der geplanten Dienstleistung relativ zu anderen Leistungen derselben Art.
- Es gibt folgende Fragen zu klären:
 - Wieviel ist der Auftraggeber bereit zu bezahlen?
 - Wieviel ist der Auftragnehmer bereit zu investieren?
 - Bis wann muss die Dienstleistung geliefert werden?
 - Welche Qualität muss die Dienstleistung haben.

IT-Projektsteuerung durch Aufträge

SE-I Vortrag von Harry Sneed aus dem Jahr 2004

- 1 Tatbestände in deutschen IT-Abteilungen
- 2 Chaos Management
- 3 Anziehende und abstoßende Kräfte in IT-Projekten
- 4 Verträge als bindende Kraft
- 5 Grund für Projektverträge
- 6 Projektergebnisse
- 7 Ergebnissteuerung in verteilten Projekten
- 8 Verträge verbinden Mitarbeiter mit Ergebnissen
- 9 Verträge regeln Ergebnisse
- 10 Vertragsverhandlungen
- 11 Spielregeln für Projektverträge
- 12 Earned Values
- 13 Verteilung der Earned Values
- 14 Ergebnisorientierte Arbeitsteilung
- 15 Auftrag für ein Fachkonzept
- 16 Auftrag für einen Komponentenentwurf
- 17 Auftrag für eine Komponente
- 18 Auftrag für eine Testprozedur
- 19 Steuerung der Mitarbeiter durch Verträge
- 20 Leistungsbezogene Vergütung

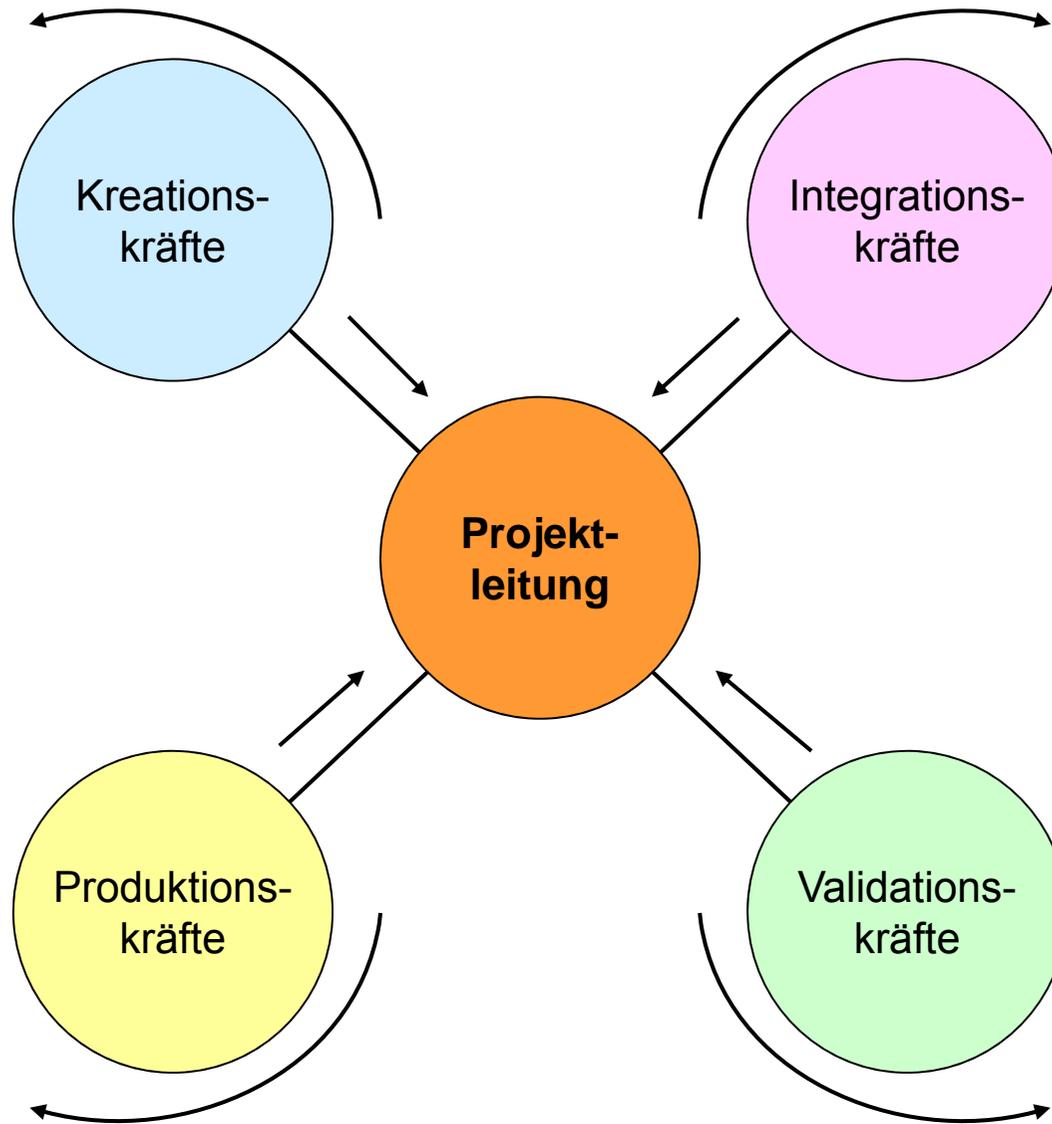
Tatbestände in deutschen IT-Abteilungen

- Ein Projekt ist ein Netzwerk einzelner Spezialisten
- jeder mit
eigenen Interessen und Zielen.
- Jedes Projektmitglied strebt nach einem
möglichst hohen Freiheitsgrad für die Erfüllung
seiner Ziele.
- Jedes Projektmitglied versucht, seine eigene
Arbeitsumgebung zu bestimmen.
- Jedes Projektmitglied möchte seinen eigenen
Weg zum Ziel finden.
- Ein autoritärer Führungsstil führt zur
Abwanderung der Projekt-
mitarbeiter bzw. zum Dienst nach Vorschrift.
- Projekte lassen sich nur über vereinbarte
Zielergebnisse führen.

Chaos Management

- Softwareentwicklung droht ständig, in Chaos auszuarten.
- Chaos entsteht durch die Abwesenheit klarer Ziele und die Anwesenheit unverträglicher Ergebnisse.
- Chaos offenbart sich in zunehmender Projektentropie.
- Projektentropie ist die Vergeudung teurer Energie, ohne dafür einen Gegenwert zu bekommen.
- Um Chaos zu verhindern, müssen klare, eindeutige und für alle unmißverständliche Ziele gesetzt werden.
- Ziele müssen wiederum in meßbare Ergebnisse umgesetzt werden.
- Chaos läßt sich über die Zielsetzung und die Ergebnisvereinbarung steuern → geordnetes Chaos.

Anziehende und abstoßende Kräfte in IT-Projekten

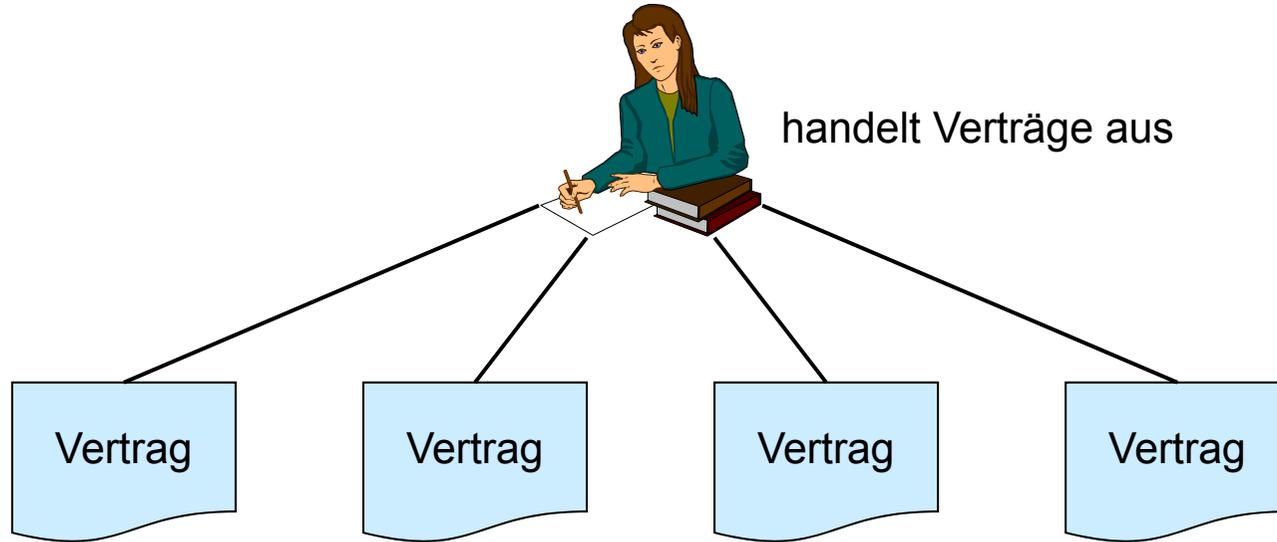


Projektleiter muss wie Schwerkraft wirken um die auseinanderstrebenden Kräfte zusammenzuhalten.

Verträge als bindende Kraft

Projektleiterin

handelt Verträge aus



Verträge sind die bindende Kraft, die das Projekt zusammenhält.



Systemarchitekt



Entwickler



Tester



Datenbank-Entwickler

Jedes Projektmitglied wird durch einen Vertrag eingebunden.

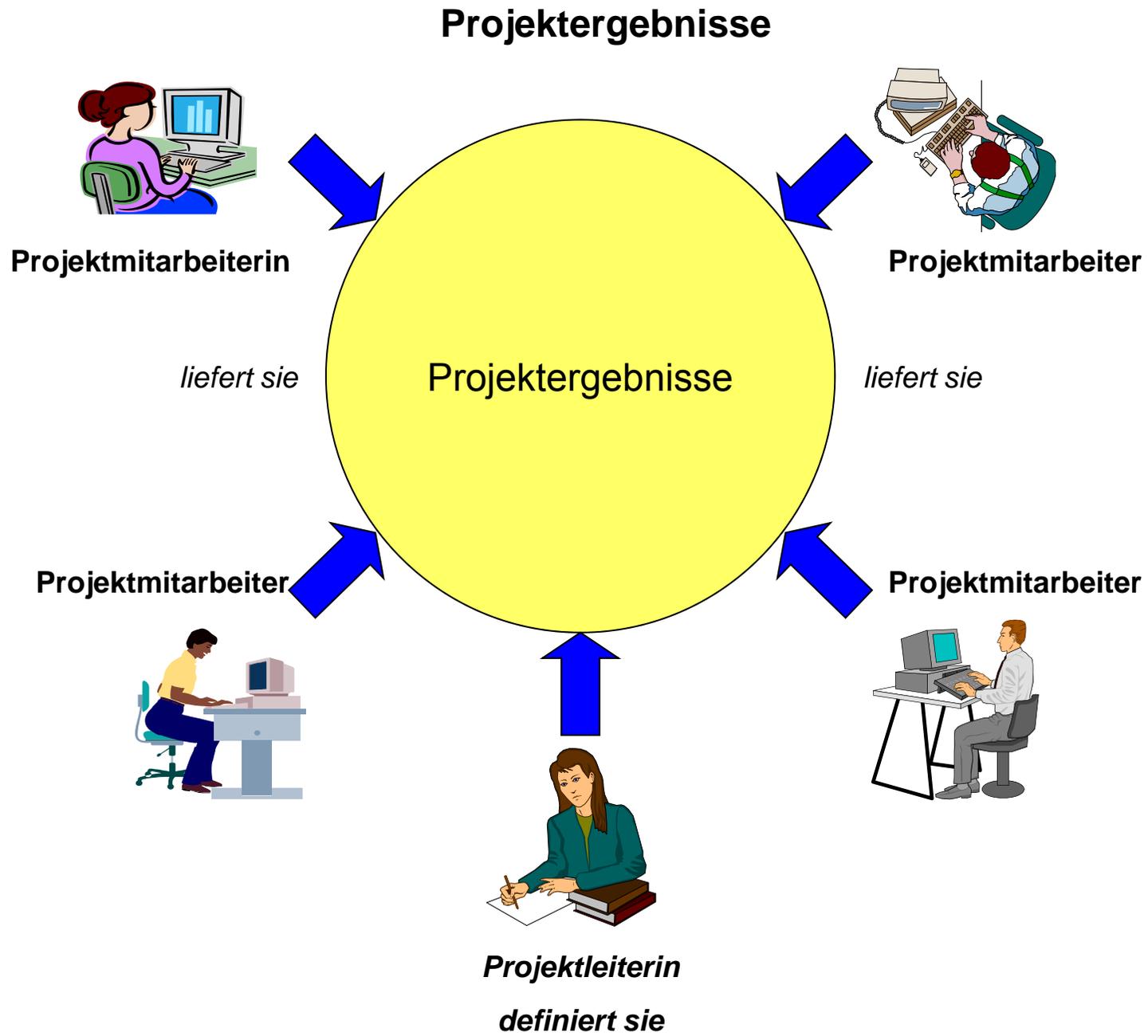
Grund für Projektverträge

**Zitat aus der
Computerwoche
Nr. 23/2001**

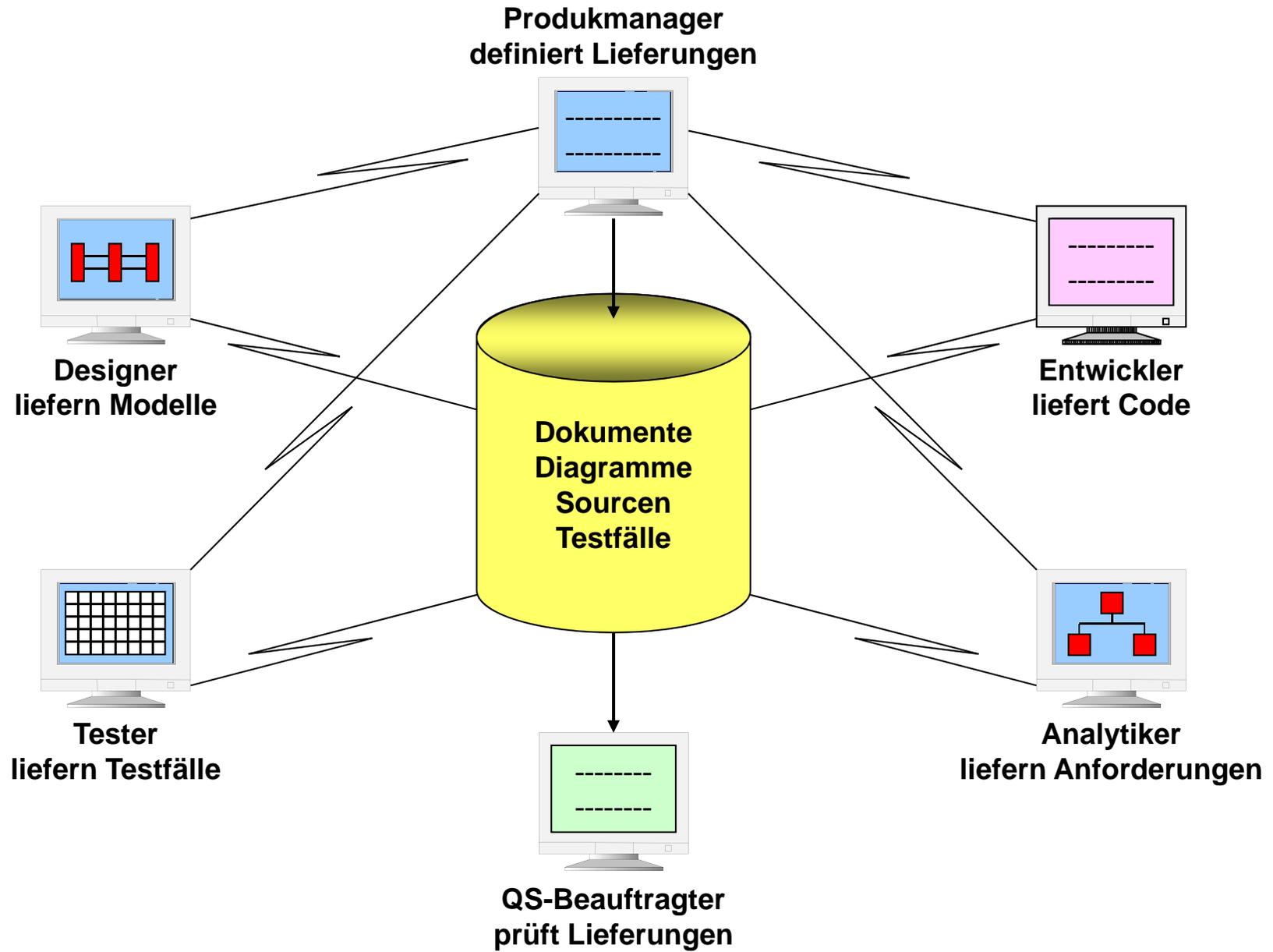
aus "Freelancing in Deutschland"

"Ein IT-Projekt ist eine einmalige Zusammenstellung ganz spezifischer technischer Anforderungen, die sich nie wiederholen. In so fern wird für jedes Projekt anderes Personal benötigt. Über die Festanstellung ist die notwendige Flexibilität nicht mehr zu erreichen. Auch die eigene technische Entwicklung dem Personalstand anzupassen und somit zu bremsen ist auf die Dauer kein sinnvoller Ausweg.

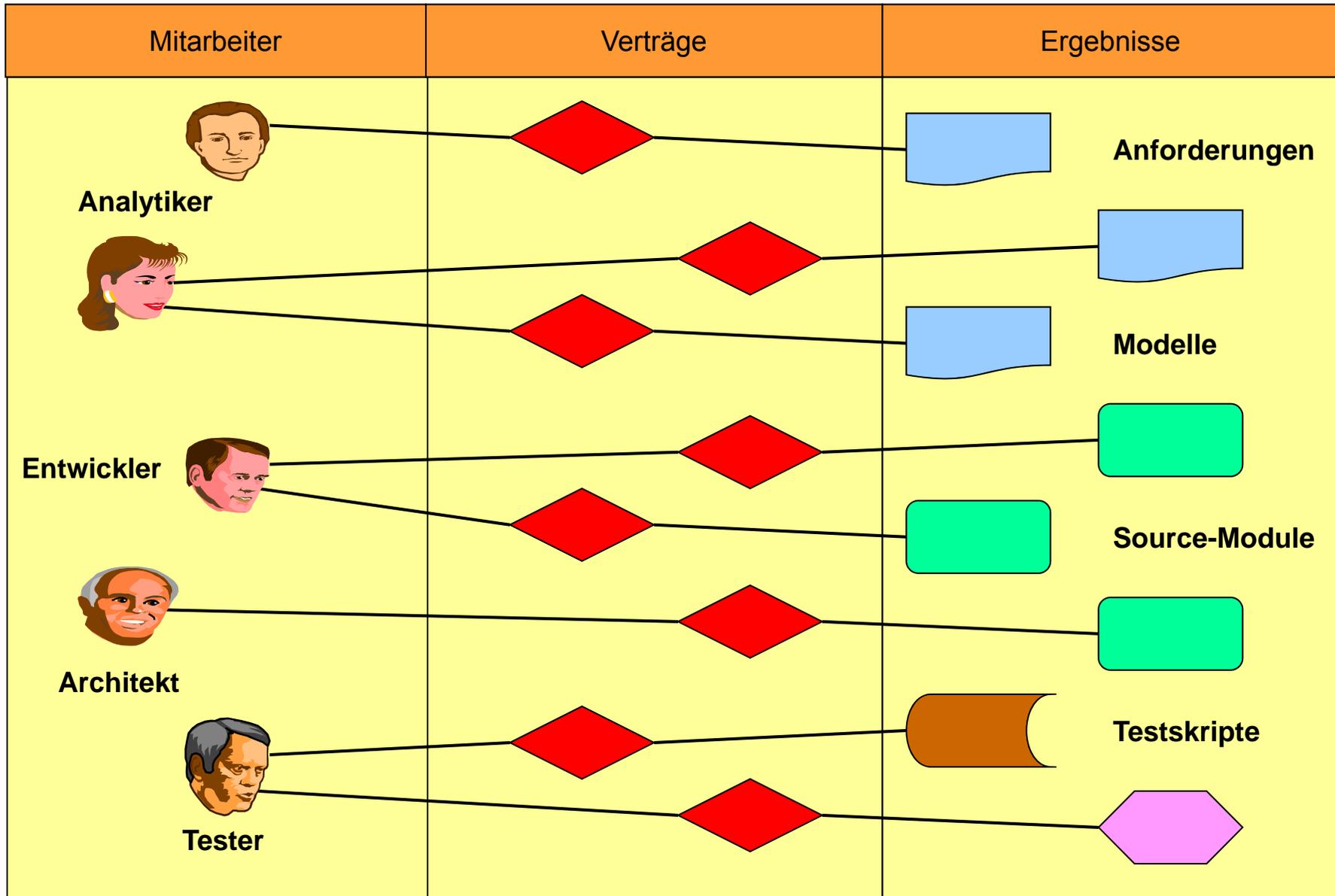
Eine bessere Lösung bietet "Freelancing", der koordinierte Einsatz von qualifizierten Freiberuflern und Selbständigen, die für jedes neue Projekt individuell zusammengestellt werden ..."



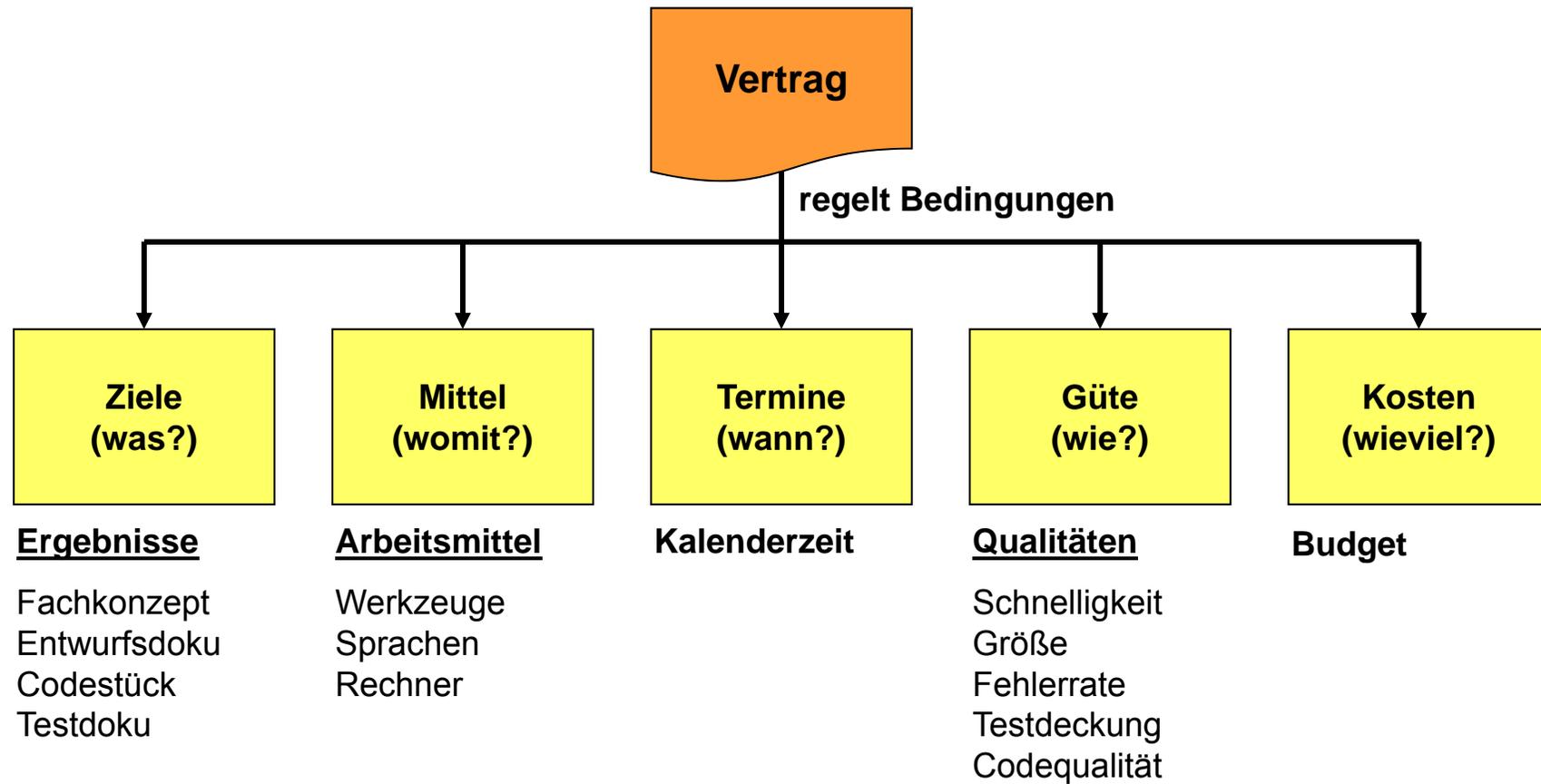
Ergebnissteuerung in verteilten Projekten



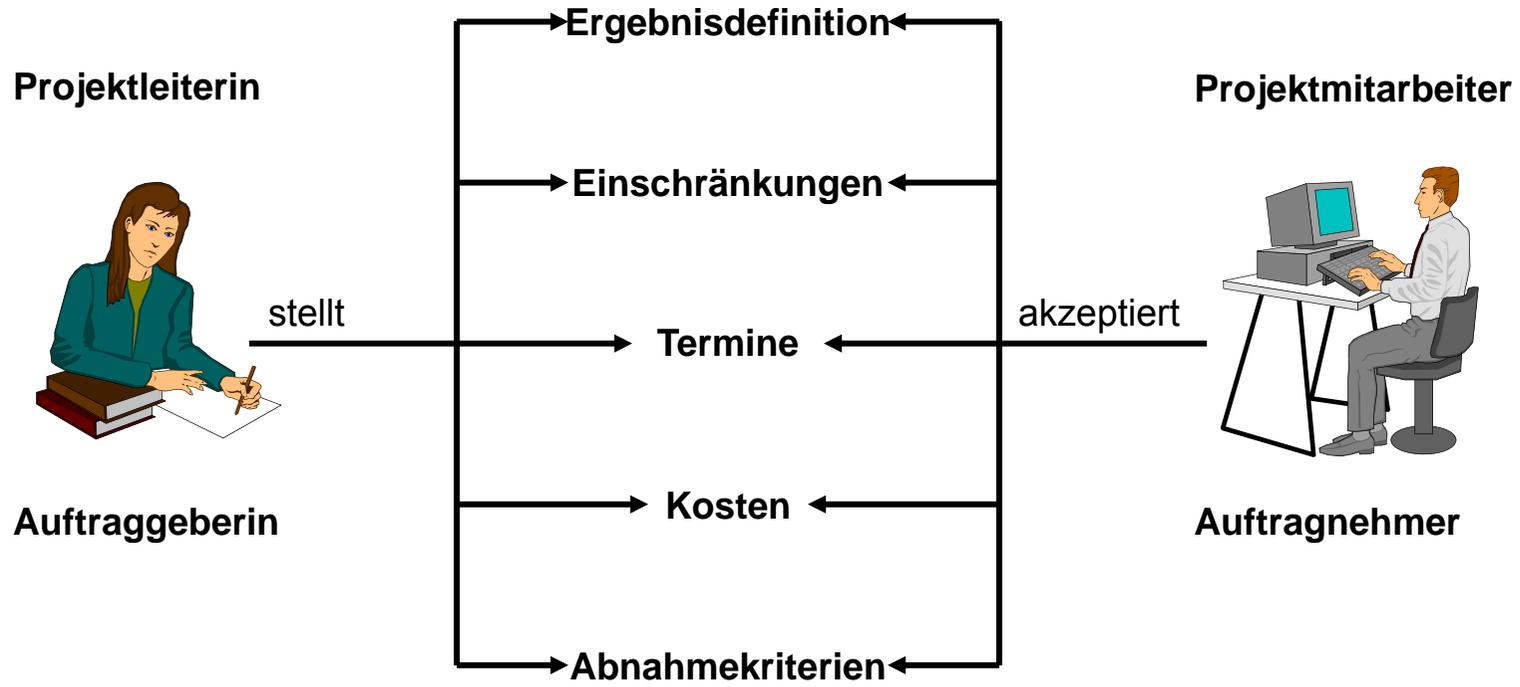
Verträge verbinden Mitarbeiter mit Ergebnissen



Verträge regeln Ergebnisse



Vertragsverhandlungen

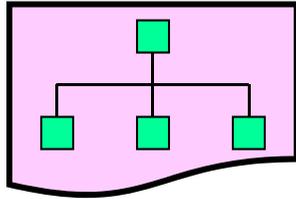


**In Deckung gehen!
Kunde droht mit
Auftrag!**

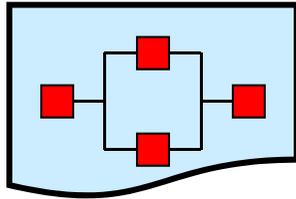
Spielregeln für Projektverträge

- Die Vertragsbedingungen werden zwischen dem Auftraggeber (Projektleiter) und dem Auftragnehmer (Projektmitarbeiter) ausgehandelt.
- Der Auftragnehmer (Projektmitarbeiter) hat das Recht, Bedingungen abzulehnen.
- Der Auftraggeber (Projektleiter) hat das Recht, seine Auftragnehmer (Projektmitarbeiter) auszusuchen.
- Niemand wird gezwungen, an einem Projekt teilzunehmen.
- Dem Projektleiter wird niemand aufgezwungen.

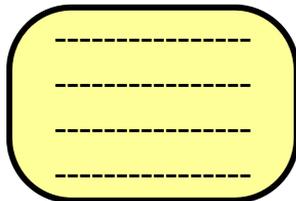
Earned Values



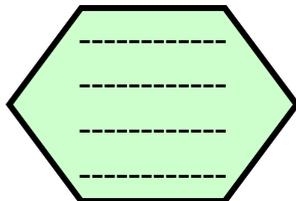
**Spezifikationsdokument = 999 Punkte pro
Anforderung &
Anwendungsfall
- Mängel**



**Entwurfsdokument = 999 Punkte pro
Diagramm
- Mängel**



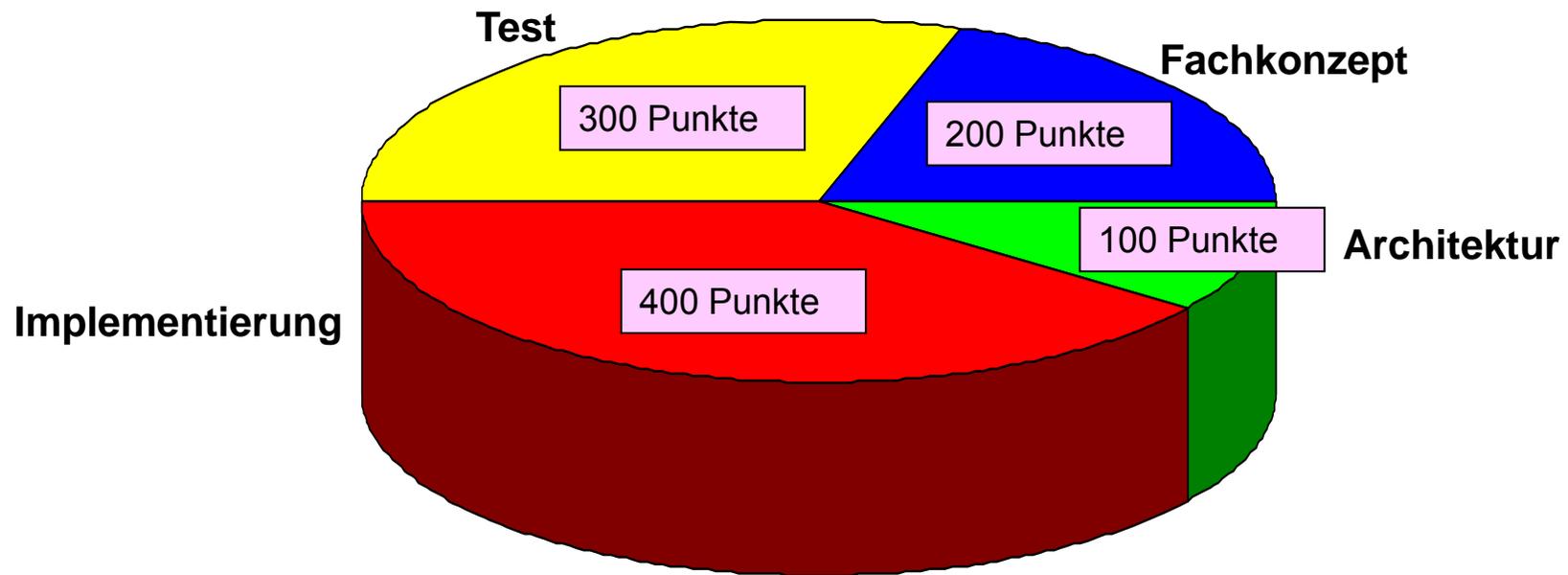
**Code-Komponente = 999 Punkte pro
Kilo Anweisungen
- Mängel
- Fehler**



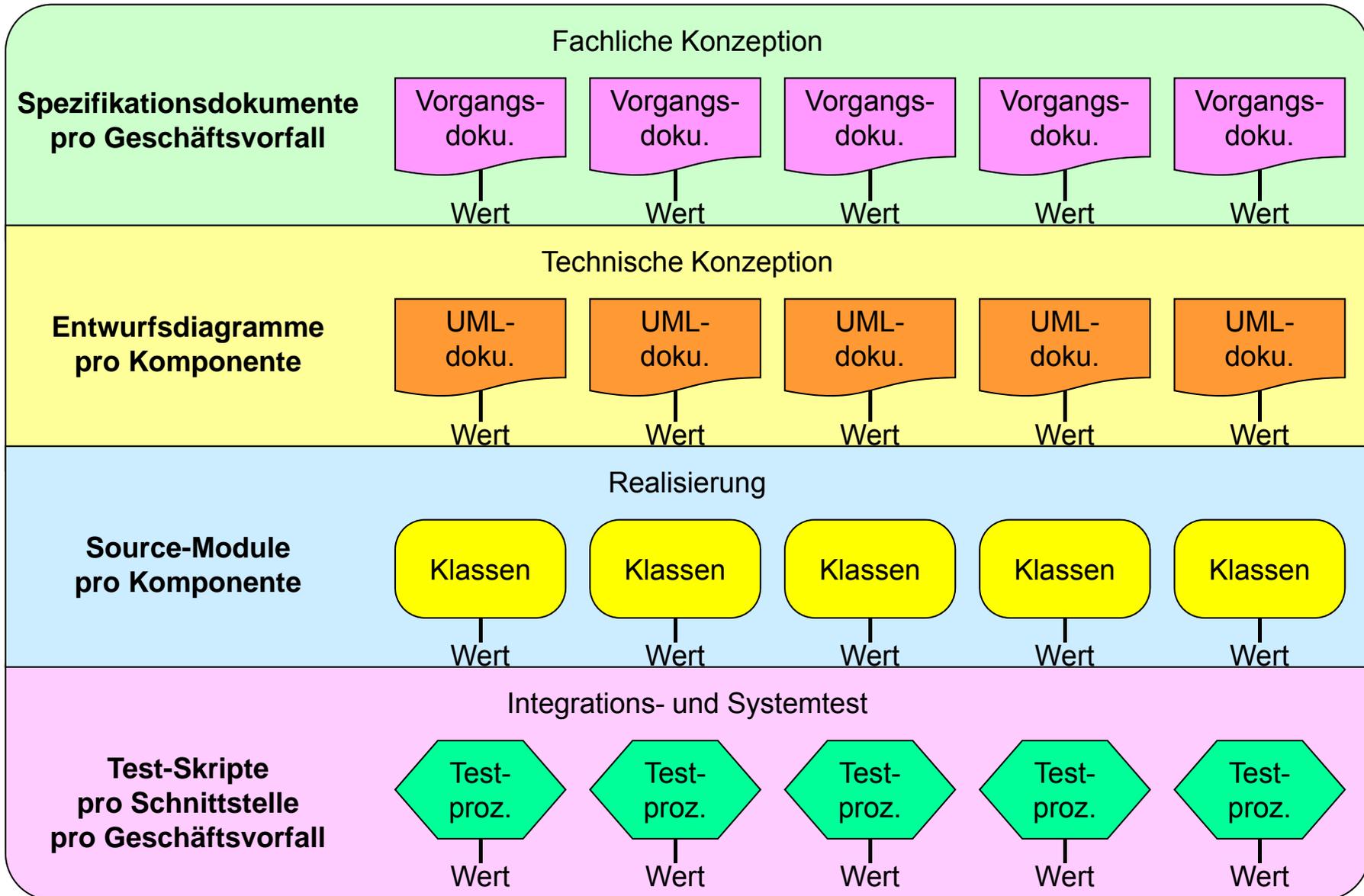
**Testprozedur = 999 Punkte pro
Kilo Testfälle
+ Fehler**

Verteilung der Earned Values

Function-Points = 1000



Ergebnisorientierte Arbeitsteilung



Auftrag für ein Fachkonzept

Ergebnis:	Requ-Diagramme für einen Geschäftsvorgang (Prozeß-Diagramm, Use Case-Diagramm, Geschäftsregel-Tabelle, Ressourcen-Tabelle, Oberflächenmuster, Drehbuch, Tabelle der fachlichen Testfälle)
Mittel:	Concept Management Facility am PC-Arbeitsplatz
Termin:	Freitag, der 13. Januar
Aufwand:	15 Personentage
Güte:	gemäß der Requirement Richtlinie <ul style="list-style-type: none">● besteht Prüfung durch CMFAnal● >90% Konformität mit den Spezifikationsregeln● muß Review durch Fachexperte bestehen
Wert:	30 Verdienstpunkte = 3 000, 00 EURO

Auftrag für ein Teilmodell

Ergebnis:	UML-Diagramme für eine Komponente (Klassen-, Sequenz-, Zustands-, Aktivitäten-Diagramme)
Mittel:	Rational Rose am PC-Arbeitsplatz
Termin:	Freitag, der 13. Februar
Aufwand:	10 Personentage
Güte:	gemäß der UML-Entwurfsrichtlinie <ul style="list-style-type: none">● besteht Prüfung durch UMLAnal● >80% Konformität mit den Entwurfsregeln● muß Review durch QS-Experte bestehen
Wert:	20 Verdienstpunkte = 2 000, 00 EURO

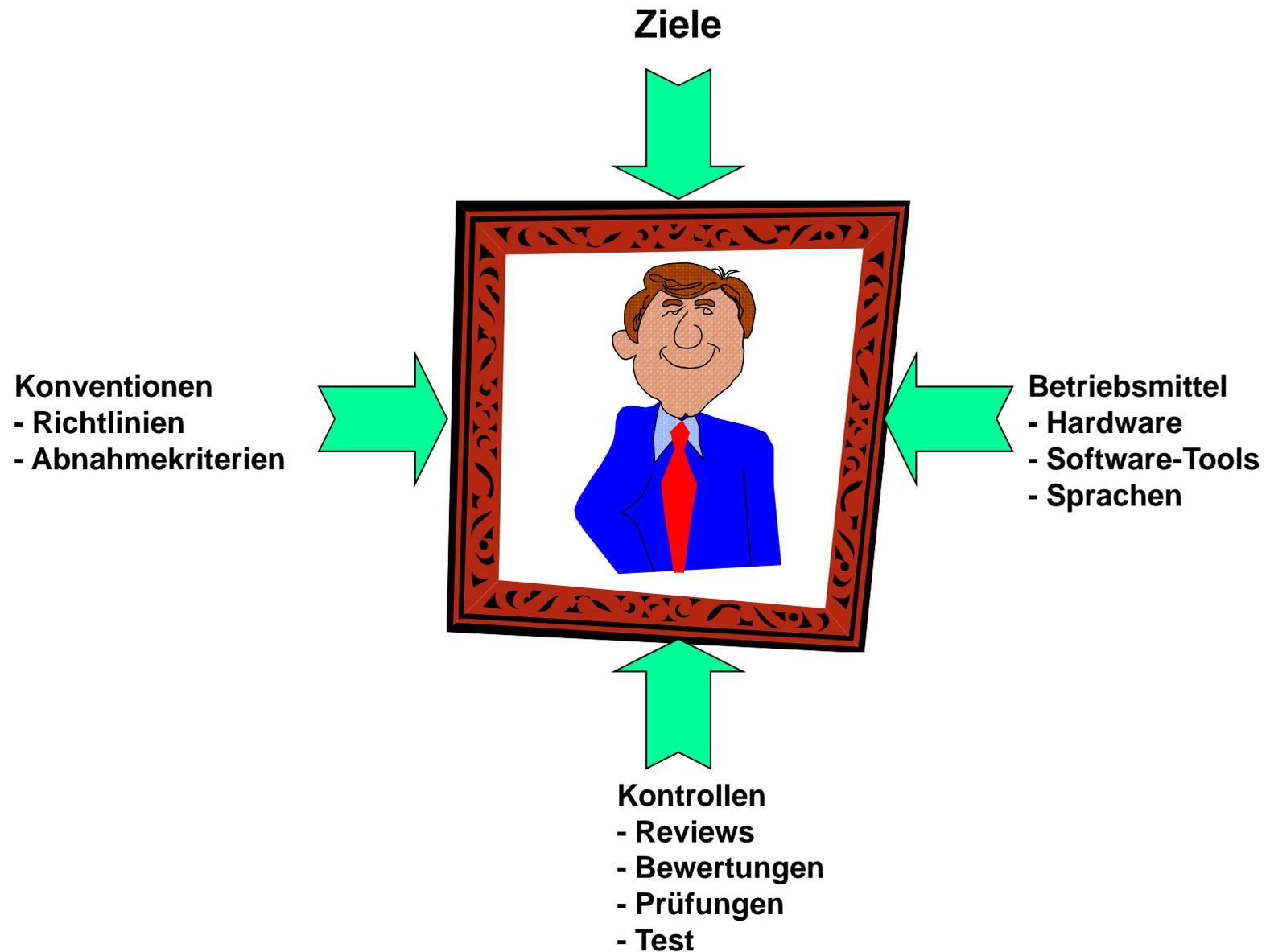
Auftrag für eine Code Komponente

Ergebnis:	Java-Klassen und -Schnittstellen SQL-Datenbankzugriffe XML-Komponentenschnittstellen
Mittel:	JBuilder, XML-Parser, Oracle
Termin:	Freitag, der 13. März
Aufwand:	20 Personentage
Güte:	gemäß der Java/XML-Programmierrichtlinie <ul style="list-style-type: none">● besteht Prüfung durch JAVAnal● >70% Konformität mit den
Wert:	Programmierregeln <ul style="list-style-type: none">● muß Codeinspektion bestehen 40 Verdienstpunkte = 4 000, 00 EURO

Auftrag für eine Testprozedur

Ergebnis:	Testfalltabelle Testskript Testüberdeckungsprotokoll Testergebnisabgleich Fehlermeldungen
Mittel:	JUnit, Java Analyzer, XML-Test, Win Runner
Termin:	Freitag, der 13. April
Aufwand:	15 Personentage
Güte:	gemäß der Testkonvention <ul style="list-style-type: none">● >75% Zweigüberdeckung● >90% Schnittstellenüberdeckung● >99% Use Case-Überdeckung● muß Testrevision bestehen
Wert:	30 Verdienstpunkte = 3 000, 00 EURO

Steuerung der Mitarbeiter durch Verträge



Leistungsbezogene Vergütung

Wer seine Verträge nicht erfüllt,
wird weniger bezahlt!

Wer seine Verträge gut erfüllt,
wird entsprechend belohnt!



**Leistungsstarke
Mitarbeiterin**



**Leistungsschwacher
Mitarbeiter**

Überleben in der GIG Welt

- Software Engineers müssen lernen in der GIG Welt zu überleben.
- Sie müssen potentielle Projekte schnell abschätzen
- Sie sollten in der Lage sein sowohl Zeit als auch Aufwand zu kalkulieren.
- Die Projektkalkulation ist das Wichtigste überhaupt.
- Ist sie zu hoch bekommt man den Auftrag nicht.
- Ist sie zu niedrig, zahlt man drauf.
- Man muss auch mit wenig Vorgaben richtig schätzen können.
- Dazu braucht man viel Erfahrung und eine umfangreiche Schätzdatenbank.

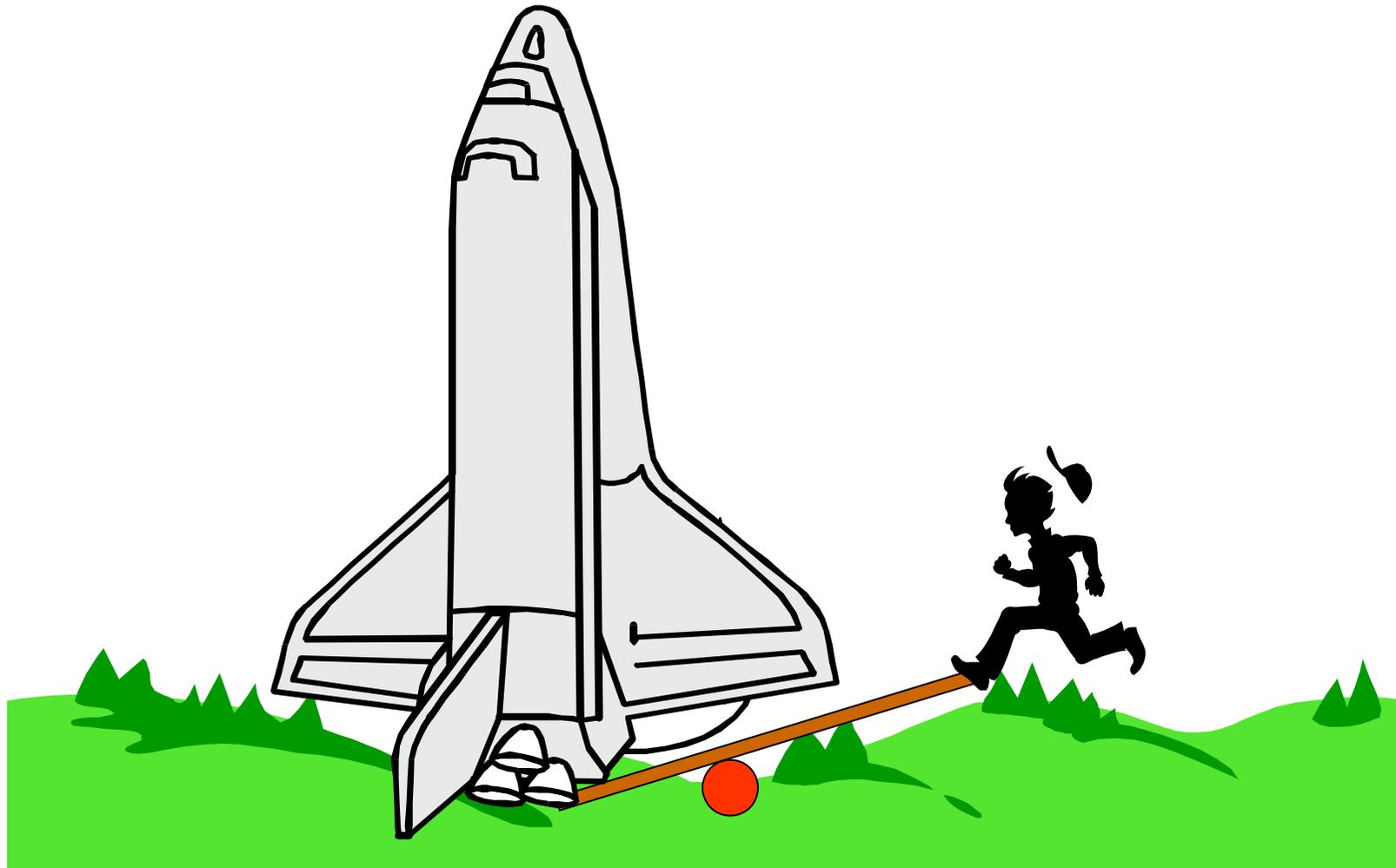
Die Preise der Dienstleistungen sind Verhandlungssache

- Software-Dienstleistungsanbieter müssen schnell reagieren können.
- Der potentielle Kunde wird gleich nachfragen, was eine bestimmte Dienstleistung kostet.
- Wie ein Call-Girl oder Call-Boy müssen sie ein Gespür dafür haben, wie viel der Kunde bereit zu bezahlen ist.
- Wenn sie zu viel verlangen, wird der Kunde gleich aufhängen, das Gespräch is beendet.
- Wenn sie zu wenig verlangen, sagt der Kunde gleich zu und beharrt auf den erstgenannten Preis.

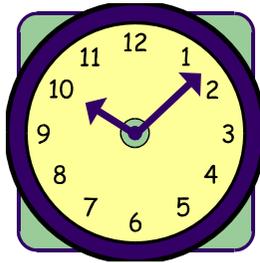
Einführung in die Schätzproblematik

- Das Verhältnis vom Zweck und Mittel
- Das Wesen eines Projektes
- Das Teufelsquadrat
- Softwaregröße
- Softwarekomplexität
- Software Qualität
- Cone of Uncertainty

Passen die Mittel zum Zweck?



Ein Software Projekt



Die Uhr
läuft

Projektziel
(6. Stunde?)

Felsen

Hier hinten
wartet das
nächste
Ziel

Gletscher

3. Meilenstein
(4. Stunde)

Steinschlag

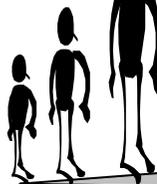
2. Meilenstein
(3. Stunde)

Mir
nach!

Plan

Berg der tausend Gefahren

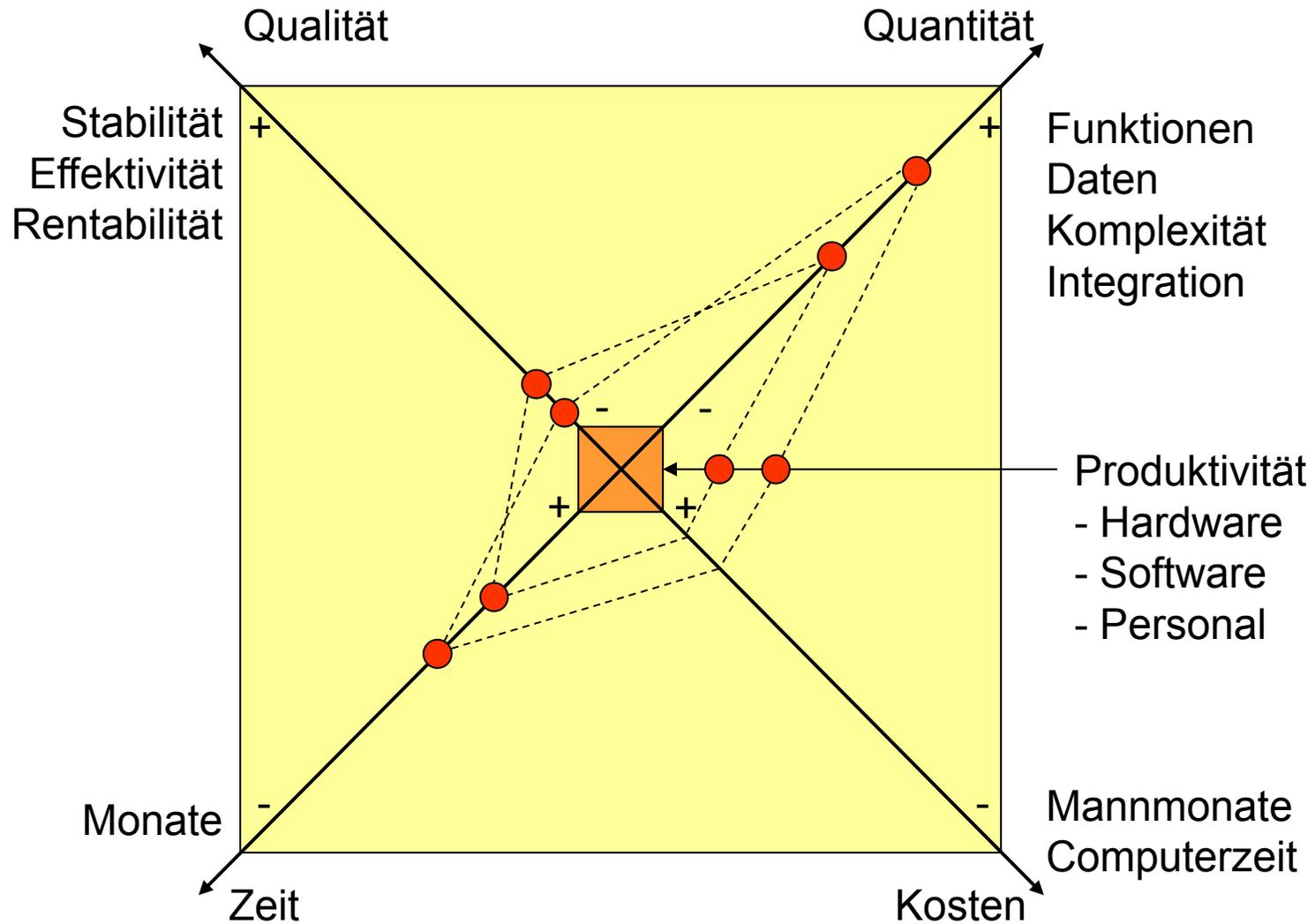
1. Meilenstein
(1. Stunde)



Projektteam

heißt ein hohes Ziel erreichen mit beschränkter Zeit und Kapazität

Das Teufelsquadrat



Software Produktivitätsmaße

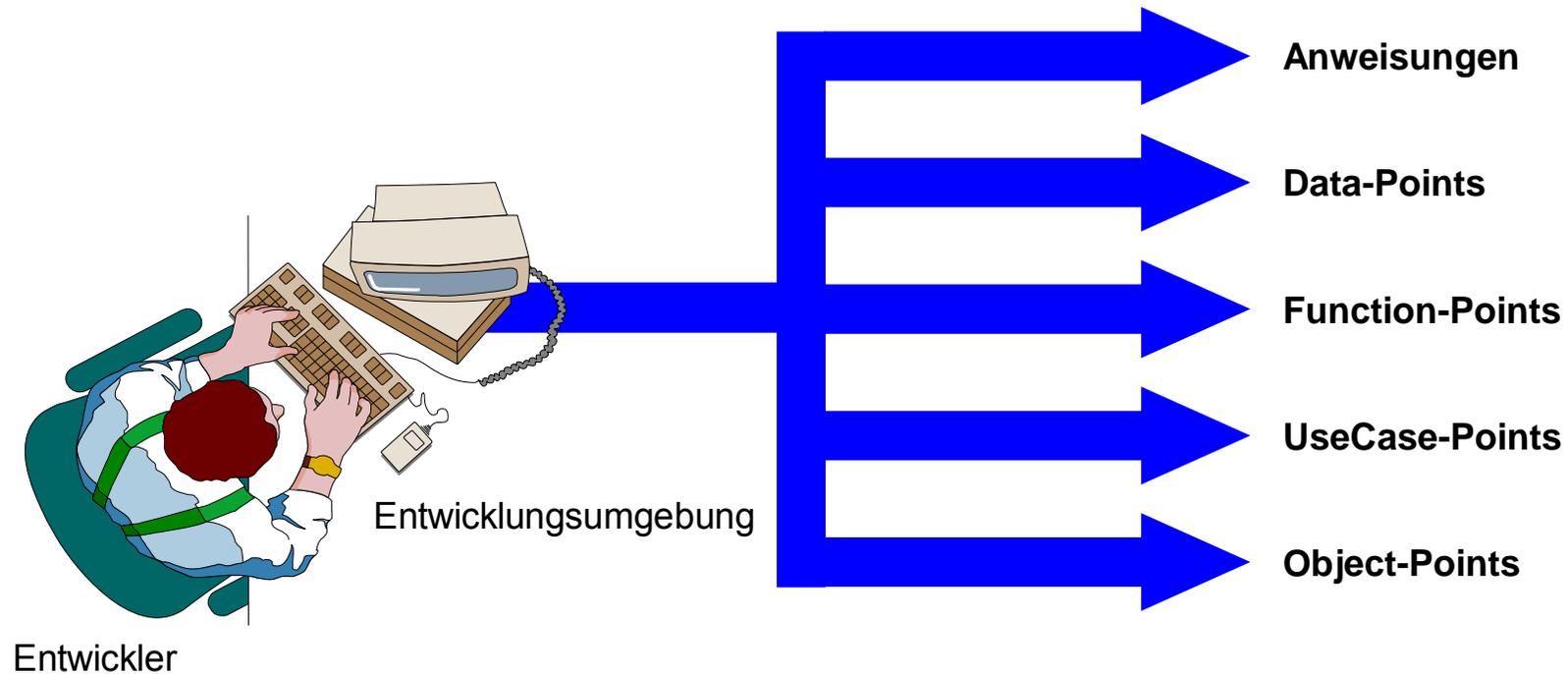
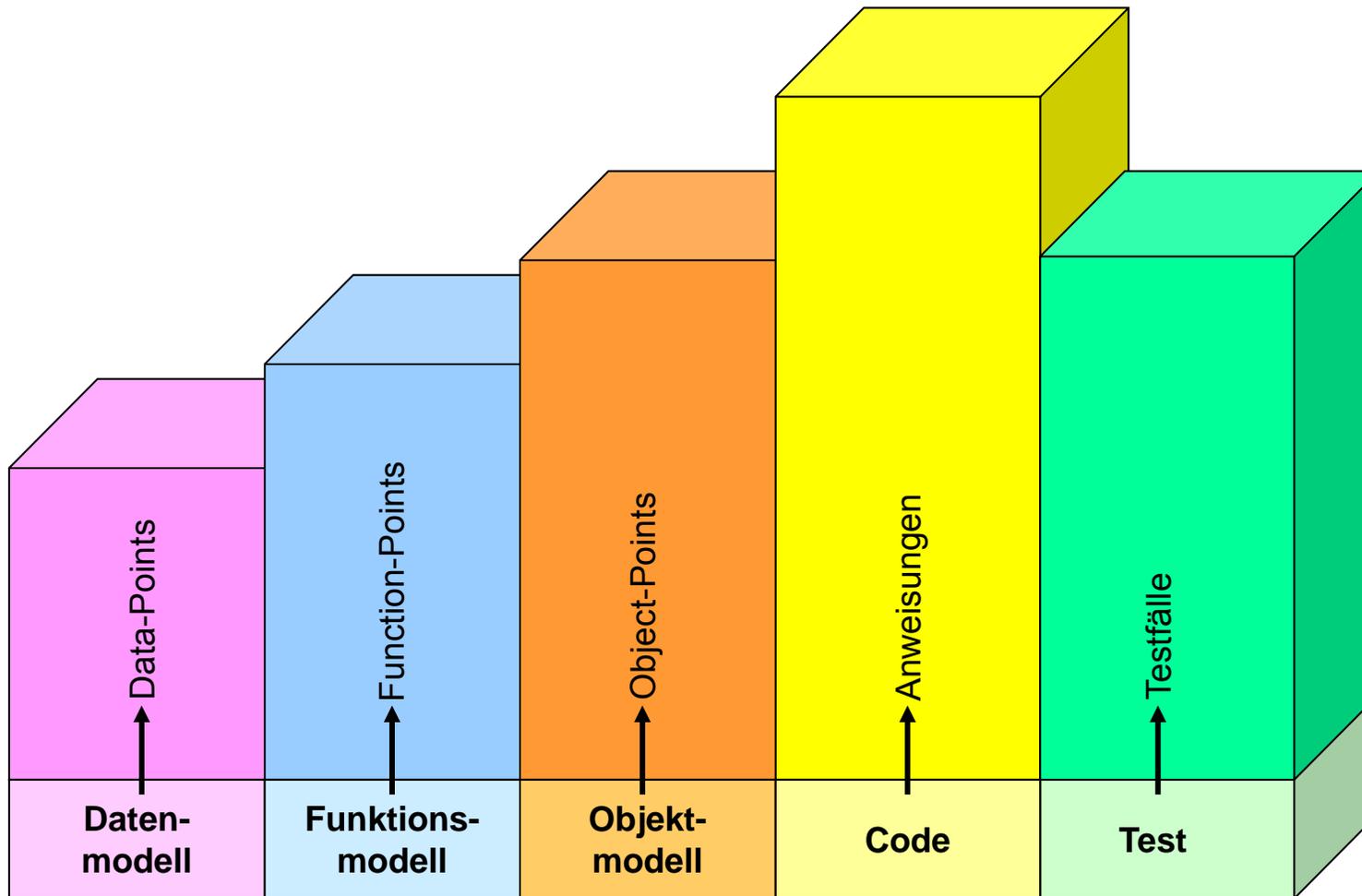
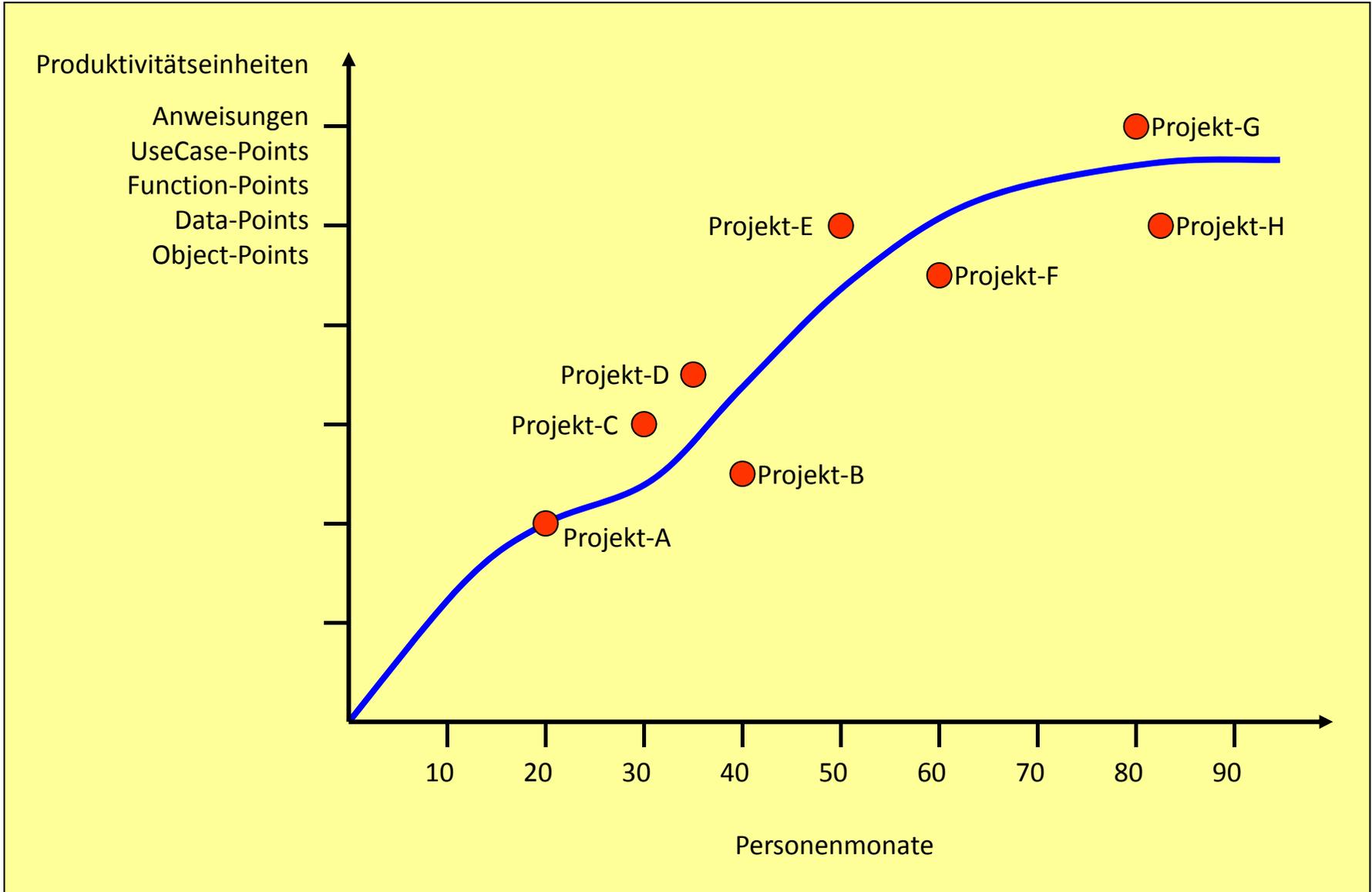


Abb. 2.1: Alternative Produktivitätseinheiten

Software-Quantitätsmaße



Produktivitätsmessung



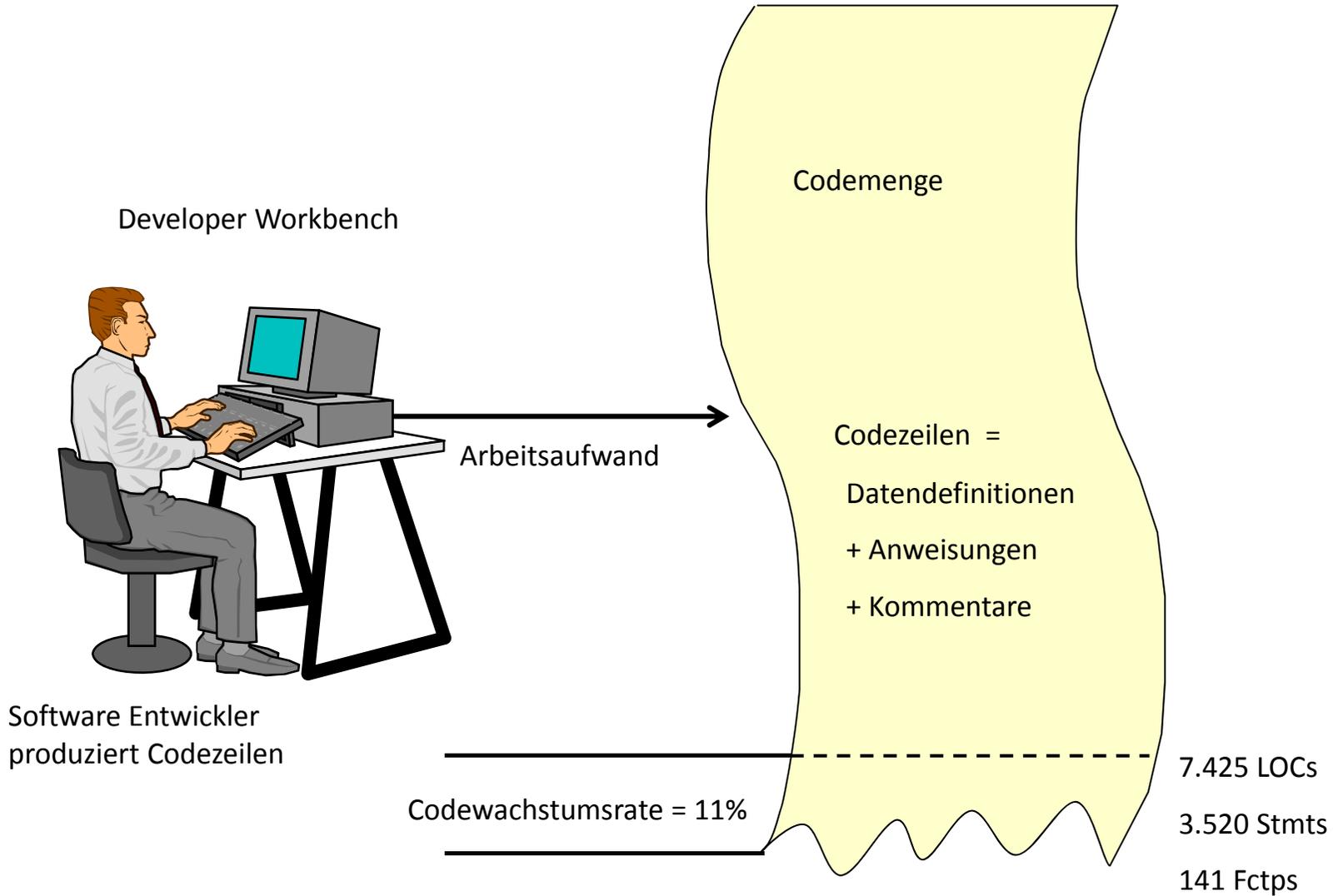
Produktivität wird anhand der Erfahrung bisheriger Projekte ermittelt

Software-Schätzverfahren

- Anweisungen, Lines of Code
- Constructive Cost Model (COCOMO)
- Object-Points
- Data-Points
- Use Case Points
- Function-Points
- Story-Points
- Testfälle bzw. Test-Points

Codegrößenmessung

Systemgröße



Das COCOMO-II Schätzmodell

Ausschlaggebend sind außer der Größe der Systemtyp, die Skalierung, die Projekteinflüsse und die angestrebte Qualität.

$$\text{Aufwand} = \text{Typ} \times \left[\frac{\text{Größe}}{\text{Produktivität}} \right]^{\text{SE}} \times (\text{EF} \times \text{QF})$$

Aufwand: = Personenmonate

Typ := Systemtyp [0,5 für Standalone
1 für Integrated
2 für Distributed
4 für Embedded Realtime]

Size := Kilo Source Anweisungen,
Function-Points, Data-Points, Object-Points oder
UseCase-Points

Prod := Anweisungen, Function-Points, Data-Points, Object-
Points, oder UseCase-Points pro Personenmonat

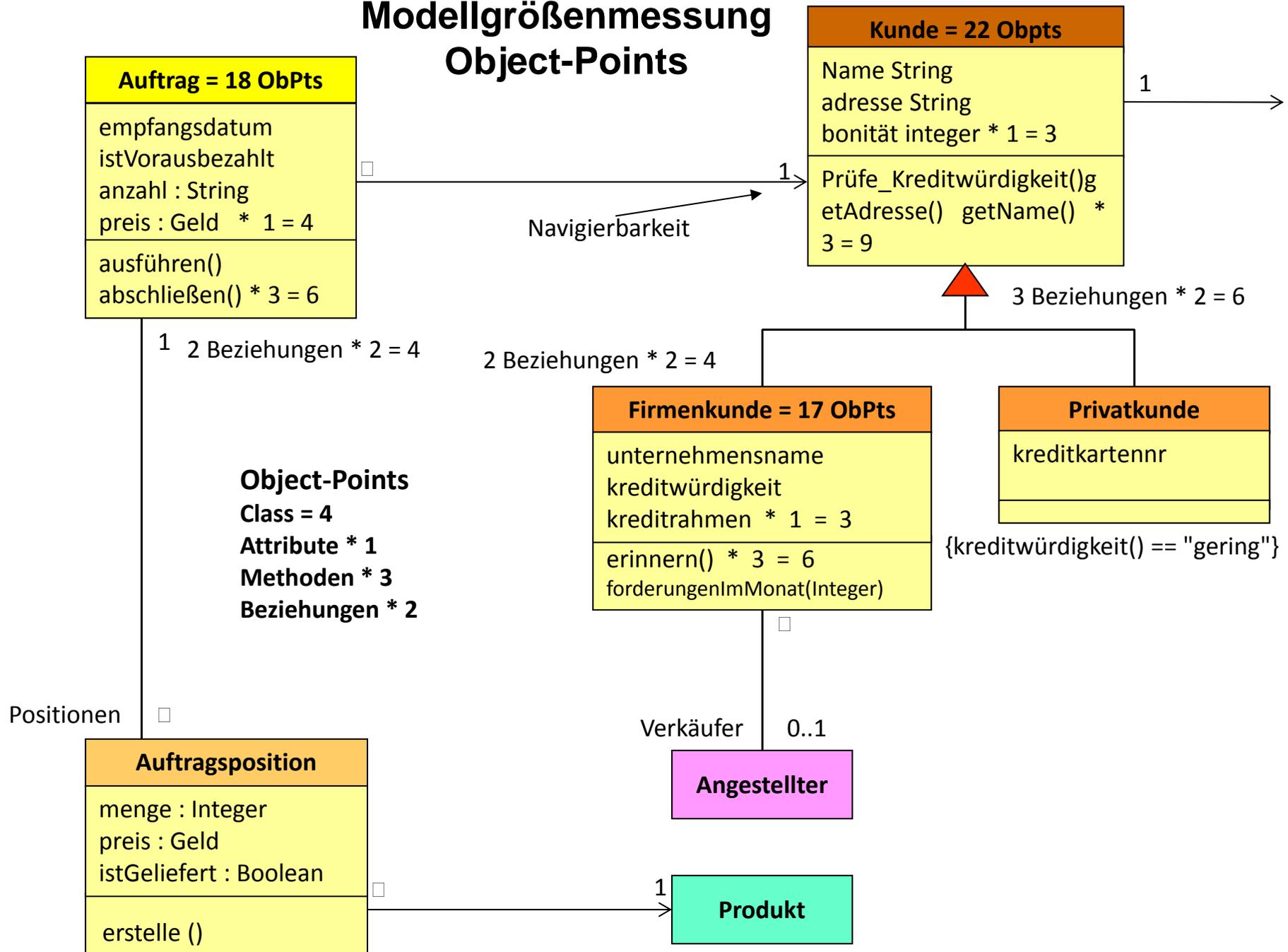
SE := Skalierungsexponent (0,91:1,23)

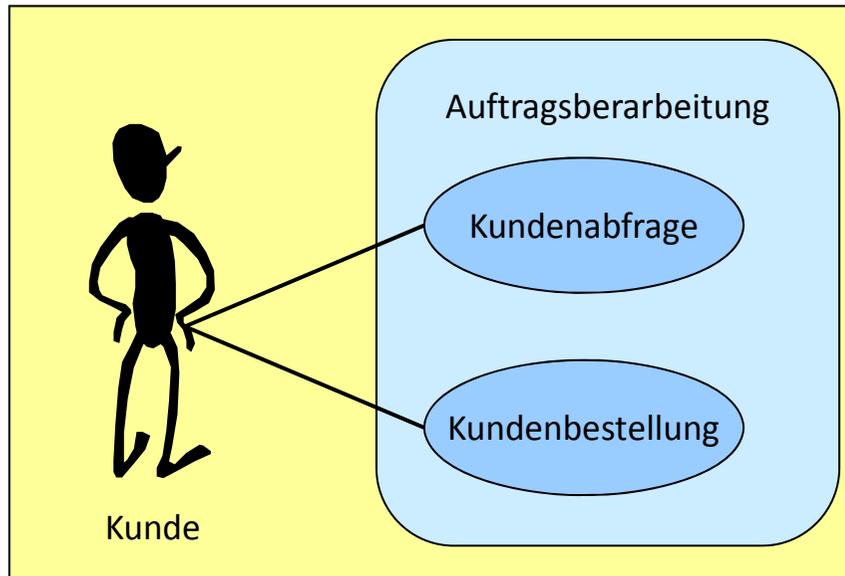
EF := Einflußfaktor (0,70:1,40)

QF := Qualitätsfaktor (0,5:1,5)

SE = Mittelwert von
Wiederverwendungsgrad,
Umgebungsreife,
Zielarchitektur,
Teamkohäsion und
Prozessreife.

Modellgrößenmessung Object-Points





Use Case Diagramm

1. Kunde selektiert den Menueintrag „Bestellung“ .
2. System zeigt Bestellungsmaske mit Artikel an.
3. Kunde wählt bis zu 10 Artikel aus und gibt die gewünschte Menge ein.
4. System prüft ob die Artikelmenge auf Lager ausreichend ist.
5. Wenn ausreichend System zieht bestellte Menge von Artikelmenge ab und erstellt Versandposten und Rechnungsposten.
6. Wenn nicht ausreichend erstellt System einen Lieferposten.

Use Case Messung

Typ	Akteur Schnittstellenart	Gewicht
Einfach	Programm Schnittstelle (API)	1
Mittel	Nachrichtenschnittstelle (XML/WSDL)	2
Komplex	Menschliche Bedienung (GUI)	3

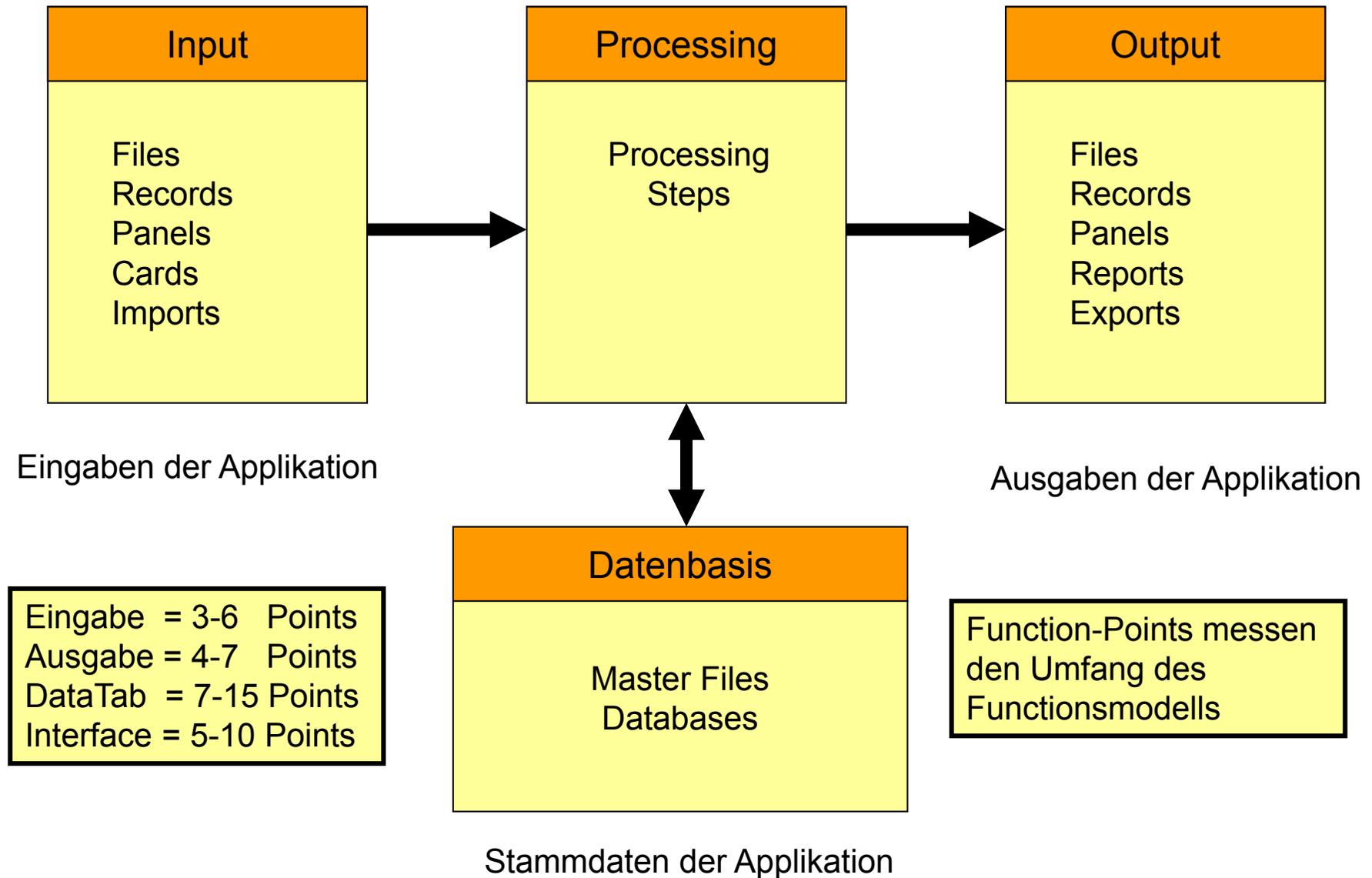
Gewichtung der Systemakteure

Typ	Anzahl der Prozessschritte
Einfach	3 oder weniger Schritte
Mittel	4 bis 7 Schritte
Komplex	Mehr als 7 Schritte

Gewichtung der Anwendungsfälle

Use Case Spezifikation

HIPO – der Ursprung der Function-Points



IFPUG-4 Function-Point Zählung

Function Type	Functional Complexity	Complexity Totals	Function Type Totals
ILF	_____ Low	x 7 = _____	
	_____ Average	x 10 = _____	
	_____ High	x 15 = _____	_____
EIF	_____ Low	x 5 = _____	
	_____ Average	x 7 = _____	
	_____ High	x 10 = _____	_____
EI	_____ Low	x 3 = _____	
	_____ Average	x 4 = _____	
	_____ High	x 6 = _____	_____
EO	_____ Low	x 4 = _____	
	_____ Average	x 5 = _____	
	_____ High	x 7 = _____	_____
EQ	_____ Low	x 3 = _____	
	_____ Average	x 4 = _____	
	_____ High	x 6 = _____	_____
Unadjusted Functional Point Count			_____

IFPUG-4 Justierungsfaktoren

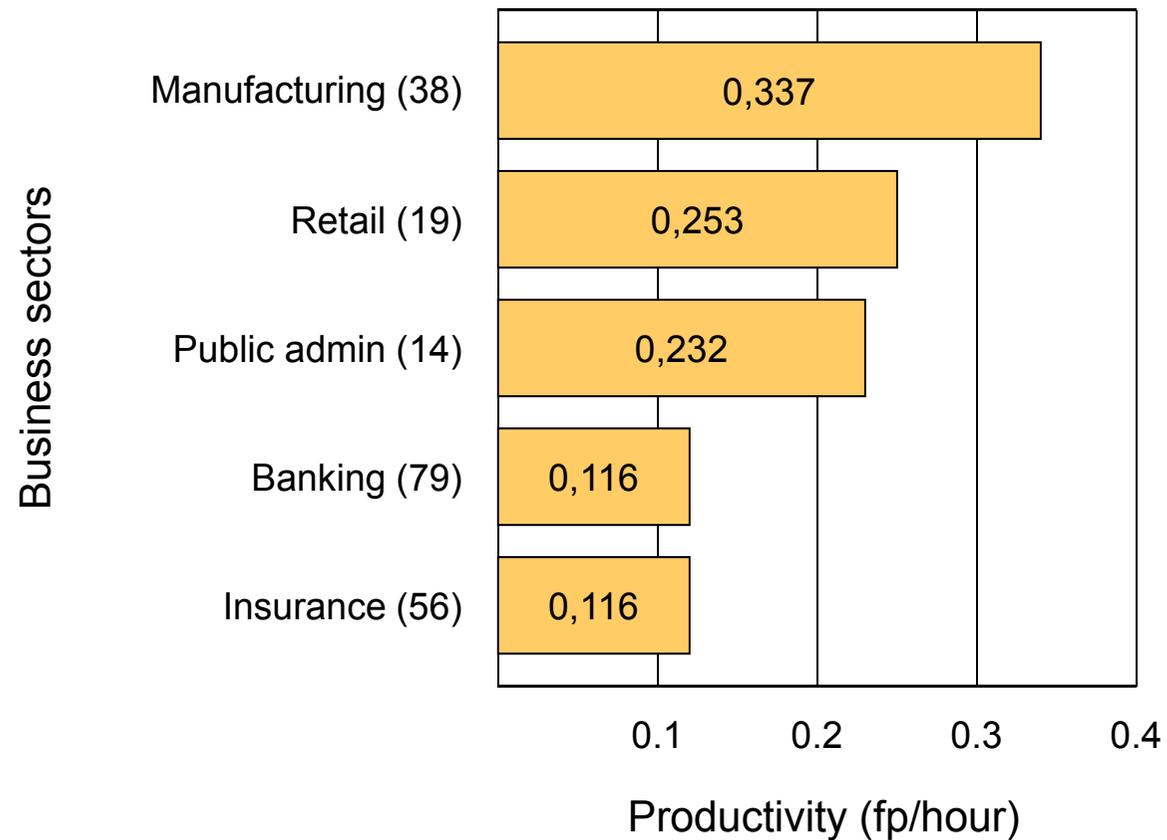
General System Characteristics (GSC)	Degree of Influence (DI) 0-5
1. Data Communication	_____
2. Distributed Processing	_____
3. Performance	_____
4. Heavily Used Configuration	_____
5. Transaction Rates	_____
6. On-Line Data Entry	_____
7. Design for End User Efficiency	_____
8. On-Line Update	_____
9. Complex Processing	_____
10. Usable in Other Applications	_____
11. Installation Ease	_____
12. Operational Ease	_____
13. Multiple Sites	_____
14. Facilitate Change	_____
Total Degree of Influences (TDI)	_____
Value Adjustment Factor (VAF) _____	= (_____ * 0.01) + 0.65
VAF	= (TDI * 0.01) + 0.65

**Function-Points per Person-Month
(Industrial Averages)
Source: David Consulting Group 2004**

Client Server Projects	17 Function-Points
Mainframe Projects	13 Function-Points
Web-based Projects	25 Function-Points
E-Business Projects	15 Function-Points
Standard Packages	18 Function-Points
Data Warehouse Projects	9 Function-Points

Ergebnis einer europäischen Produktivitätsstudie

() = Anzahl Projekte untersucht



Quelle: Kathrin Maxwell, IEEE Software, 2000

Aufwandsschätzung einer Bankapplikation

- FIDIS Bankapplikation
- Automatische Messung des Fachkonzeptes
- Projektion der Spezifikation auf die Implementierung
- Automatische Messung des Source-Codes
- Schlüsse aus der Schätzung

Function-Point Schätzung auf Basis des Systementwurfs

	KMS			PMS		GEA	KOS		ZAS				FIS	Zentrale Systeme				Total
	KMS	PVBE	MMS	PMS	PMSI	GEA	KOS	ITM	ZAS	ARPL	AUF	TRM	FIS	KOND	ARS	LAW	STS	
Batch Proc.	0	0	0	0	0	0	546	0	117	39	0	117	195	0	0	0	78	1092
Dialoge Proc.	2418	182	0	78	572	1300	1274	78	234	494	104	104	494	1066	286	546	1300	10530
Service Proc.	1354	414	0	1587	161	2553	4554	138	460	1058	184	391	2047	552	1265	92	739	17549
Data Objects	570	150	0	120	0	250	790	10	30	100	50	120	550	110	500	0	50	3400
Reports	0	117	0	0	0	39	0	0	39	0	0	0	195	0	0	0	0	390
Anz. Tabellen	915	120	0	300	0	390	465	30	0	195	225	90	195	630	360	300	1350	5565
Total	5260	983	0	2085	733	4532	7629	256	880	1886	563	822	3676	2358	2411	938	3514	38526
Person Months	263	49,2	0	104,3	36,7	226,6	381,5	12,8	44,0	94,3	28,2	41,1	183,8	117,9	120,6	46,9	175,7	1926,3

Subsystem	FPs	PM's	PY's
KMS	6243	312	26
PMS	2818	141	12
GEA	4532	227	19
KOS	7885	394	33
ZAS	4151	208	17
FIS	3676	184	15
Zentrale Systeme	9221	461	38
Gesamt	38526	1926	161

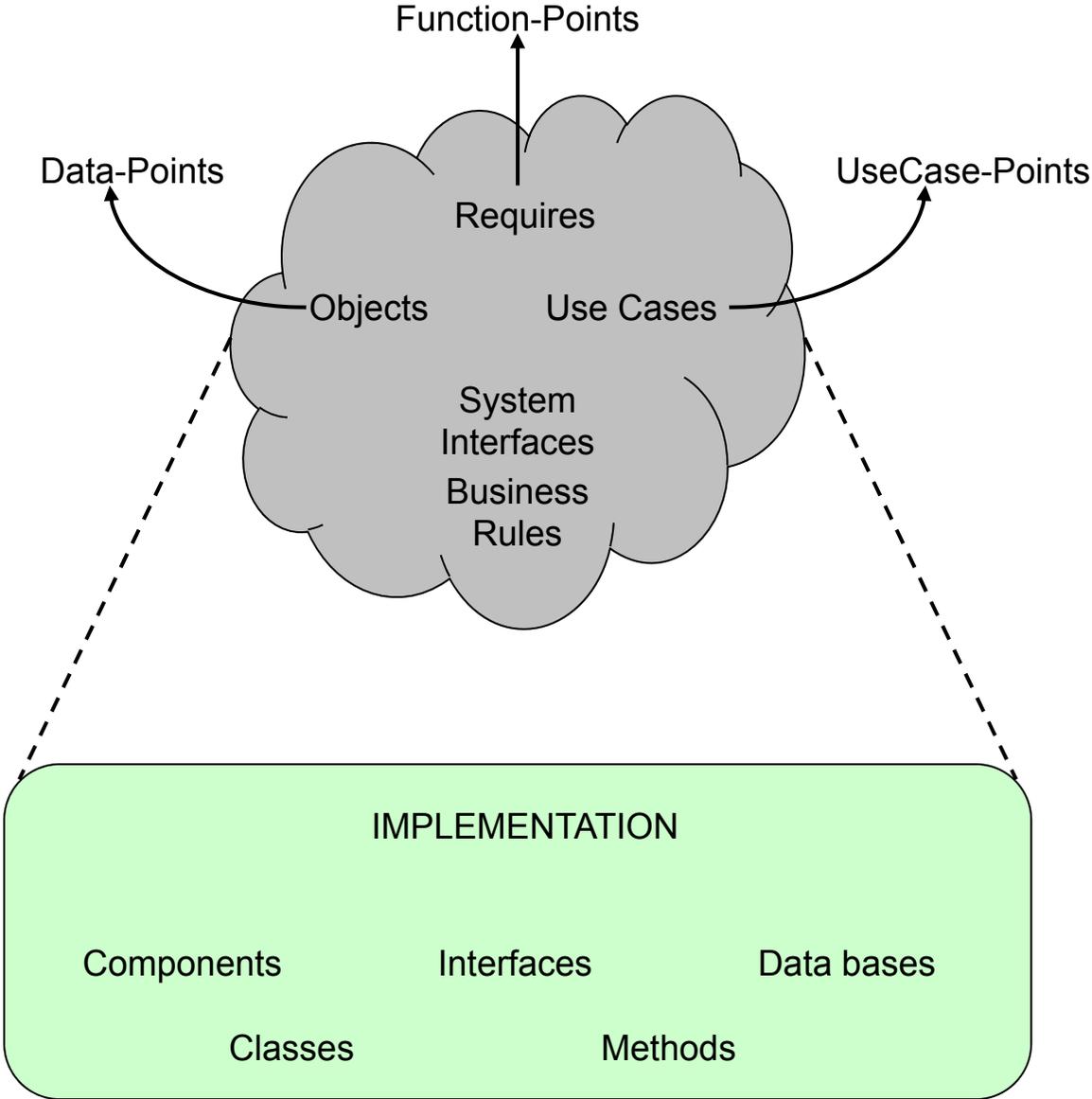
Estimated Development
Costs of Drive = 1926 PM

Actual Development
Costs of Drive = 1756 PM

Type	Berechnung	Faktor
Batch Process	(3 Inputs x 6)+(3 Outputs x 7)	39
Dialog Process	(2 Inputs x 6)+(2 Outputs x 7)	26
Service Process	(2 Inputs x 4)+(3 Outputs x 5)	23
Data Objects	(x 10)	10
Tables	(x 15)	15
Reports (Müller)	wie Batch	39

Person month (PM) = 20 PDs
Person Year (PY) = 11 PMs

Projektion der Anforderungen auf die Implementierung



Automatisierte Anforderungsmessung

```
-----+-----+
          C M F A N A L   C O N C E P T   M E T R I C   R E P O R T
LANGUGAGE: Objekt                                DATE: 25.03.99
PROCESS:d:\fidis\sts\FIDIS                       PAGE:      1
-----+-----+
```

```
          C O N C E P T   Q U A N T I T Y   M E T R I C S
Number of Concept Members analyzed              =====>    3171
Number of Concept Lines ina all                 =====> 2470490
Number of Structure Concept Lines              =====>   253135
Number of Text Lines                           =====> 2217355
```

```
          S T R U C T U R A L   Q U A N T I T Y   M E T R I C S
Number of Processes analyzed                   =====>    1305
Number of Online Processes analyzed            =====>     379
Number of Object Frame Processes analyzed      =====>     52
Number of Service Processes analyzed          =====>     41
Number of Batch Processes analyzed            =====>      3
Number of Structures analyzed                 =====>     97
Number of Data Structures included             =====>   1323
Number of Text Members referenced             =====>  21182
Number of Functions referenced                =====>   3840
Number of Maps sepcified                     =====>   1014
Number of Reports specified                   =====>     30
Number of Views specified                    =====>   1845
Number of Keys specified                     =====>   3690
Number of Parameters specified               =====>   1705
Number of Rules specified                    =====>    462
Number of Actions specified                  =====>   1322
Number of Objects specified                  =====>   2271
Number of Controls specified                 =====>   5543
Number of Procedures specified               =====>    958
Number of Atributes specified                =====>  11498
Number of Map Fields specified               =====>  10757
```



Automatisierte Anforderungsmessung



D I A L O G Q U A N T I T Y M E T R I C S		
Number of Dialog Menues	=====>	56
Number of Dialog Editbars	=====>	56
Number of Dialog Folders	=====>	56
Number of Dialog Events	=====>	319
T E X T Q U A N T I T Y M E T R I C S		
Number of Sentences contained in texts	=====>	20608
Number of Conditions contained in texts	=====>	4916
Number of Nouns contained in texts	=====>	115764
Number of Constants contained in texts	=====>	26005
S Y S T E M S I Z E M E T R I C S		
Number of Function-Points	=====>	52258
Number of Data-Points	=====>	178222
Number of Object-Points	=====>	193991
C O N C E P T Q U A L I T Y M E T R I C S		
Number of Changes to the Concept	=====>	3878
Number of Major Rule Violations	=====>	20
Number of Medium Rule Violations	=====>	1913
Number of Minor Rule Violations	=====>	14291
Number of Rule Violations in Total	=====>	16224
Degree of Conformance	=====>	0.935
Degree of Complexity	=====>	0.263
Quality Rating	=====>	0.928

Automatisierte Source-Code Messung

S O U R C E
LANGUAGE: C++ DATE: 10.08.01
SYSTEM: DRIVE PAGE: 1

Q U A N T I T Y M E T R I C S

C O D E	Q U A N T I T Y	M E T R I C S
Number of Source Members analyzed	=====>	4463
Number of Source Lines in all	=====>	1480242
Number of Genuine Code Lines	=====>	820807
Number of Comment Lines	=====>	413753
Number of Major Rule Violations	=====>	68753
Number of Medium Rule Violations	=====>	26802
Number of Minor Rule Violations	=====>	107379

S T R U C T U R A L Q U A N T I T Y M E T R I C S

Number of Modules	=====>	947
Number of Includes	=====>	16992
Number of Classes declared	=====>	1747
Number of Classes inherited	=====>	1718
Number of Methods declared	=====>	13410
Number of Methods inherited	=====>	8865
Number of Procedures declared	=====>	16749
Number of Interfaces declared	=====>	26343
Number of Object-Points	=====>	230809

D A T A Q U A N T I T Y M E T R I C S

Number of Panels processed	=====>	624
Number of Reports produced	=====>	54
Number of Files declared	=====>	1106
Number of Data Bases accessed	=====>	526
Number of Data Views selected	=====>	2024
Number of Data Structures	=====>	5891
Number of Defined Definitions	=====>	25810
Number of Data Variables declared	=====>	87646
Number of Data Variables inherited	=====>	21161
Number of Data-Points	=====>	226559

Source Code Metrik

PROCEDURAL QUANTITY METRICS

Number of Statements	=====>	513443
Number of Input Operations	=====>	2442
Number of Output Operations	=====>	5763
Number of File & Database Accesses	=====>	4612
Number of Function References	=====>	174136
Number of Foreign Functions referenced	=====>	97284
Number of Macro References	=====>	58754
Number of Macros referenced	=====>	11709
Number of If Statements	=====>	53398
Number of Switch Statements	=====>	3760
Number of Case Statements	=====>	20871
Number of Loop Statements	=====>	8008
Number of GOTO Branches	=====>	0
Number of Return statements	=====>	38968
Number of Control Flow Branches	=====>	182651
Number of all Control Statements	=====>	93186
Number of Literals in Statements	=====>	77308
Number of Nesting Levels (Maximum)	=====>	14
Number of Test Cases (Minimum)	=====>	116546
Number of different Statement Types	=====>	260101
Number of Assertions made	=====>	192
Number of Function-Points	=====>	48485

Aufwandsschätzung einer Auftragsbearbeitung

- Anwendungsgeschichte
- Akteure und Anwendungsfälle
- Story-Points
- Automatisierte UML Messung
- PC-Calc Schätzung
- SoftCalc Schätzung

Eine Anwendungsgeschichte

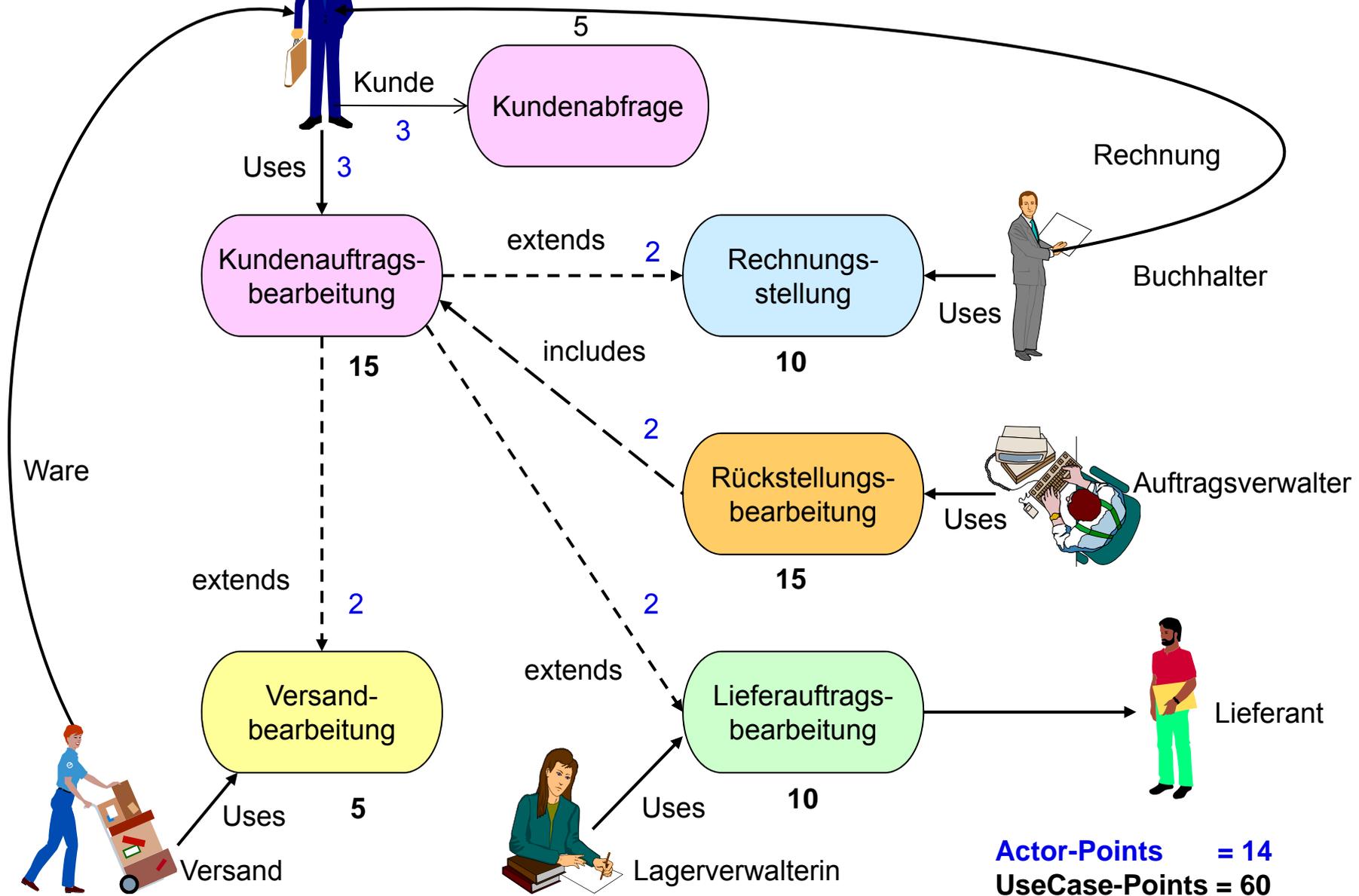
Das Auftragsbearbeitungssystem bearbeitet die Aufträge, die von den Kunden kommen. Sie werden entweder erfüllt, abgelehnt oder zurückgestellt. Wenn ein Auftrag eintrifft, wird geprüft ob der Kunde bekannt ist und ob seine Bonität in Ordnung ist. Falls der Kunde unbekannt oder seine Bonität nicht in Ordnung ist, wird der Auftrag abgelehnt.

Jeder Kundenauftrag kann bis zu 10 Bestellposten enthalten. Jeder Bestellposten hat die Nummer, der Name und die Bestellmenge des gewünschten Artikels. Erst, wird geprüft ob der gewünschte Artikel auf Lager ist. Wenn nicht, wird diese Bestellung abgelehnt. Als nächstes wird geprüft ob die Artikelmenge auf Lager ausreichend ist, um die Bestellung zu erfüllen. Wenn nicht, wird aus dem Bestellposten eine Rückstellung erzeugt. Falls, der Artikel vorhanden und die Menge ausreichend ist, wird die bestellte Menge abgezogen, die Auslieferung angewiesen und ein Rechnungsposten erstellt.

Immer wenn die Artikelmenge auf Lager reduziert wird, soll geprüft werden, ob die Menge noch über die Mindestmenge ist. Wenn nicht, ist eine Lieferbestellung für den im Moment billigsten Lieferant automatisch zu erzeugen. Ein Lieferauftrag wird aus den Lieferbestellungen eines Lieferanten gebildet und am Ende jeder Woche an die betroffenen Lieferanten versendet.

Die Rechnungsposten sind am Ende jeder Woche zu sammeln und nach Kunde zu ordnen. Jeder Kunde bekommt eine Rechnung für die von ihm in der Woche bestellten Artikel. Der Stückpreis der einzelnen Artikel wird mit der Menge multipliziert, um den Rechnungsbetrag zu ermitteln. Dazu kommt noch der im Moment gültigen Mehrwertsteuersatz. Rechnungen sind innerhalb von 30 Tage zu begleichen.

Akteure & Anwendungsfälle

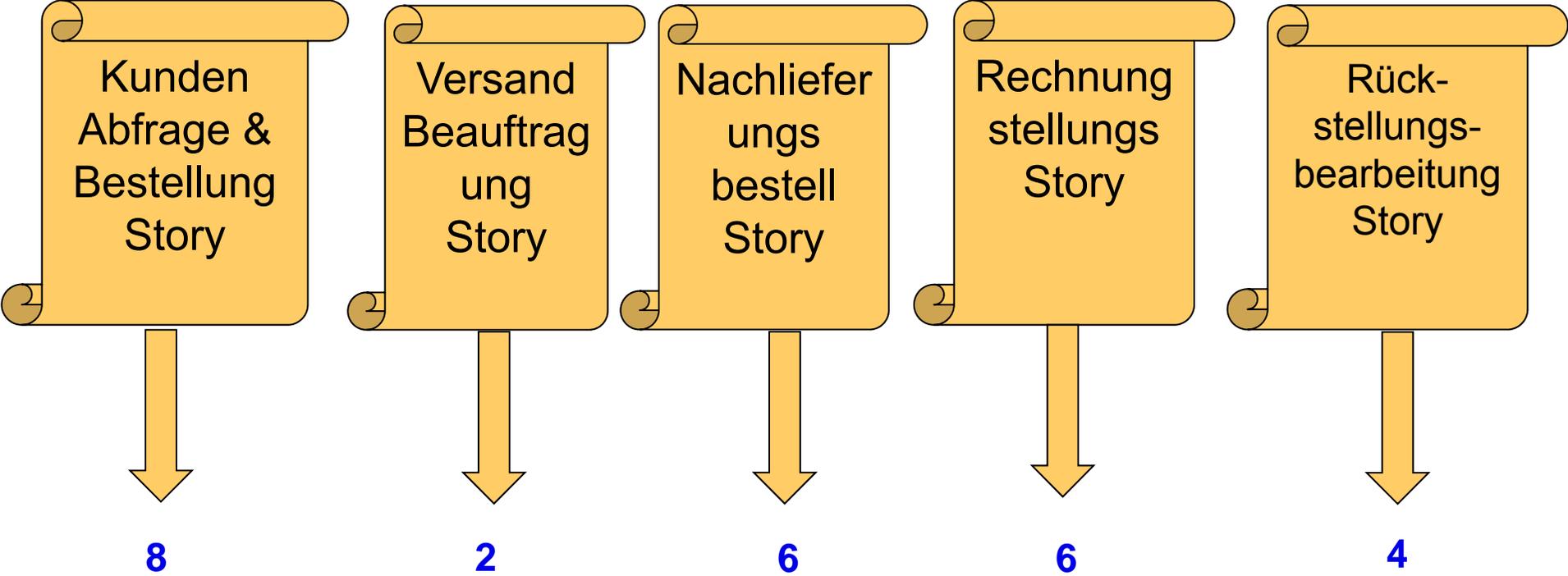


Actor-Points = 14
UseCase-Points = 60

Story-Point Zählung



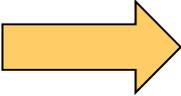
Product Owner erzählt Stories



26 Story-Points +



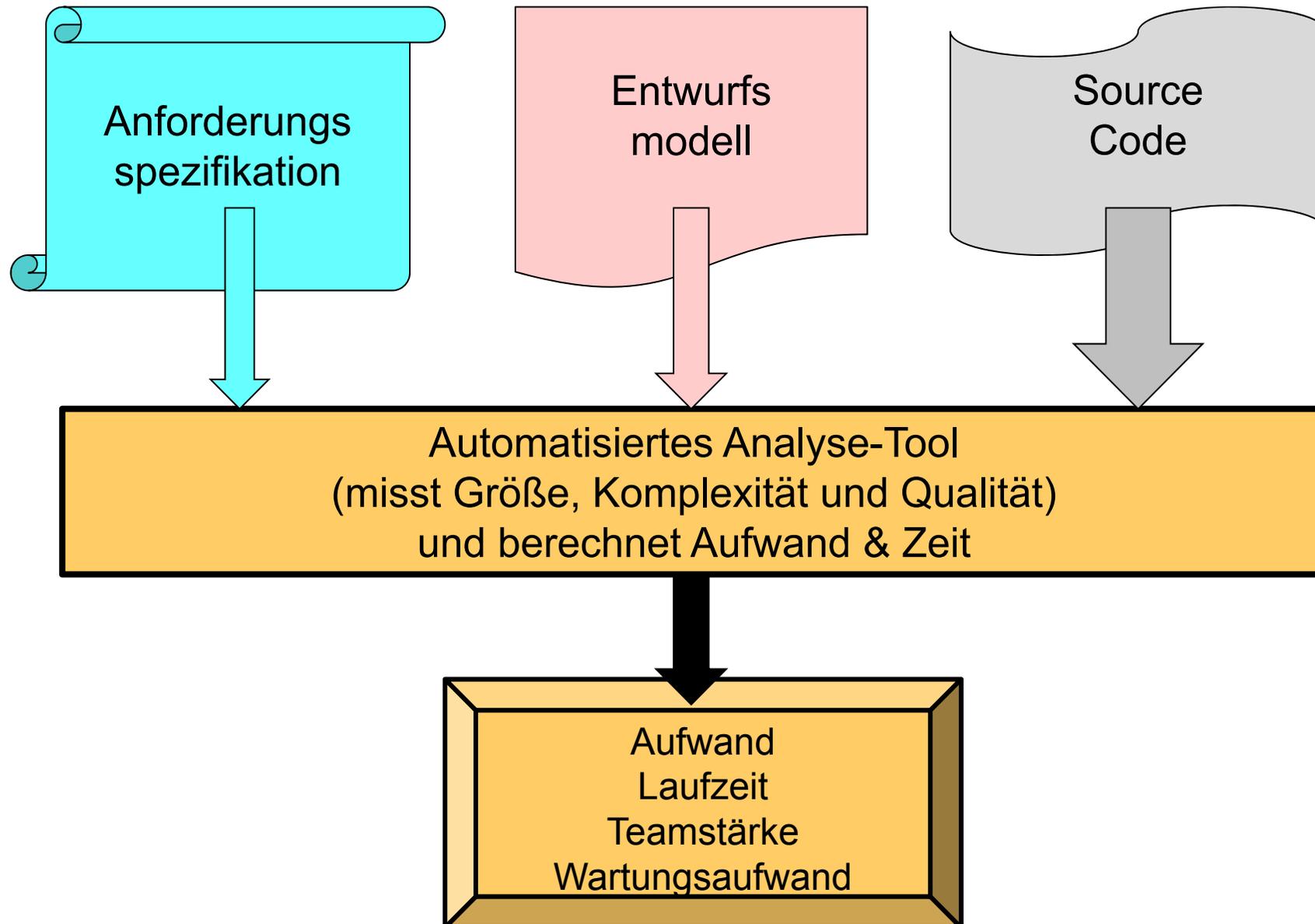
Tester



8 Test-Points = 34 Story-Points

/ 3 Points per PM = 11 PMs

Automatisierte Aufwandsschätzung



Automatisierte Aufwandschätzung mit SoftCalc

<p><u>System quantities:</u></p> <p>UseCases: 6 Objects: 12 Interfaces: 12 Components: Test Cases:</p>	<p><u>Effort estimates</u></p> <p>Undjusted Effort: 5,8 PMs <input checked="" type="checkbox"/> Influence Factor 1,07 Influence adjusted Effort: 6,3 PMs <input checked="" type="checkbox"/> Resource Factor 1,00 Resource adjusted Effort: 6,3 PMs <input checked="" type="checkbox"/> Risk Factor 1,36 Risk adjusted Effort: 8,5 PMs <input checked="" type="checkbox"/> Use Overhead Factor 0,10 Final Effort: 9,4 PMs</p>
<p><u>Size measurement</u></p> <p>Unadjusted Size: 155,00 Complexity Factor: Complexity adjusted Size: Quality Factor: 1,00 Quality adjusted Size: 155,00 Final adjusted Size: 170,50</p>	<p><u>Project estimates</u></p> <p>Minimum Effort: 8,5 PMs Minimum Time: 5,1 Months Minimum Cost: 42.570 Optimal Staff: 1,7 Prs. Annual Maint. Effort: 1,0 PMs</p>

Zusammenfassung

- Um Projekte zu kalkulieren, muß man die Produktivität der Mitarbeiter, die Größe und Komplexität des Vorhabens und die Projektbedingungen kennen.
- Produktivität wird anhand bisheriger Projekte ermittelt.
- Man muß die Projekttypen unterscheiden und je nach Projekttyp den richtigen Größenmaß auswählen.
- Man soll mit mindestens drei Schätzmethoden schätzen.
- Man muß den geschätzten Aufwand immer mit den Erfahrungswerten vergleichen.
- Man sollte die Schätzparameter immer wieder kalibrieren, denn Schätzen ist ein Lernprozeß, der nie zu Ende geht.