

Teil V: Projektmanagement

50

Prof. Dr. rer. nat. Uwe Aßmann

Institut für Software- und
Multimediatechnik

Lehrstuhl Softwaretechnologie

Fakultät für Informatik

TU Dresden

Version 16-0.1, 09.07.16

- 1) Projektmanagement
- 2) Vorgehensmodelle



DRESDEN
concept
Exzellenz aus
Wissenschaft
und Kultur

SalesPoint Informationen


Alles zu SalesPoint und zum Praktikum unter :

<http://www.salespoint-framework.org/>

http://www.st.inf.tu-dresden.de/SalesPoint/v5.0/wiki/index.php/Main_Page

- ▶ Bestandteile der Dokumentation von SalesPoint sind:
 - Überblick und Einstieg
 - Technischer Überblick zu SalesPoint
 - Tutorial zu einem Anwendungsbeispiel (FastFood-Restaurant)
 - API-Spezifikation der Framework-Klassen (javadoc) einschließlich der Beschreibung des Anpassungsinterfaces
- ▶ Dokumentation zahlreicher studentischer Praktikumsprojekte
- ▶ Infos zur Praktikumsdurchführung

50.1 Projektmanagement



und eines
Projekt

Das Glück des Lebens besteht nicht darin, wenig oder keine Schwierigkeiten zu haben, sondern sie alle siegreich und glorreich zu überwinden.

Carl Hilty, 28.02.1833 - 12.10.1909

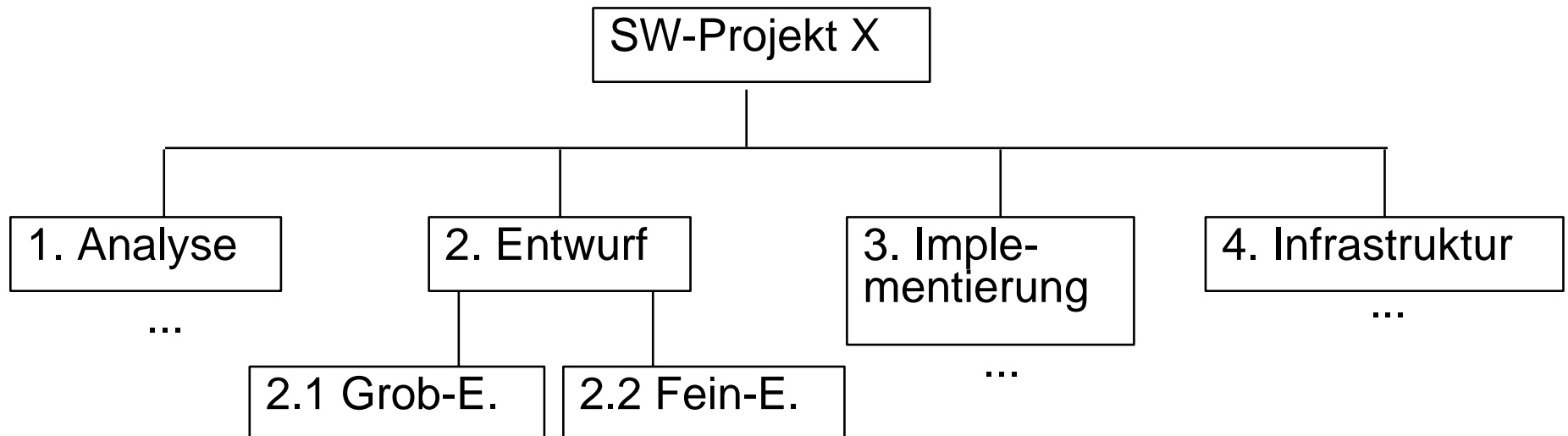
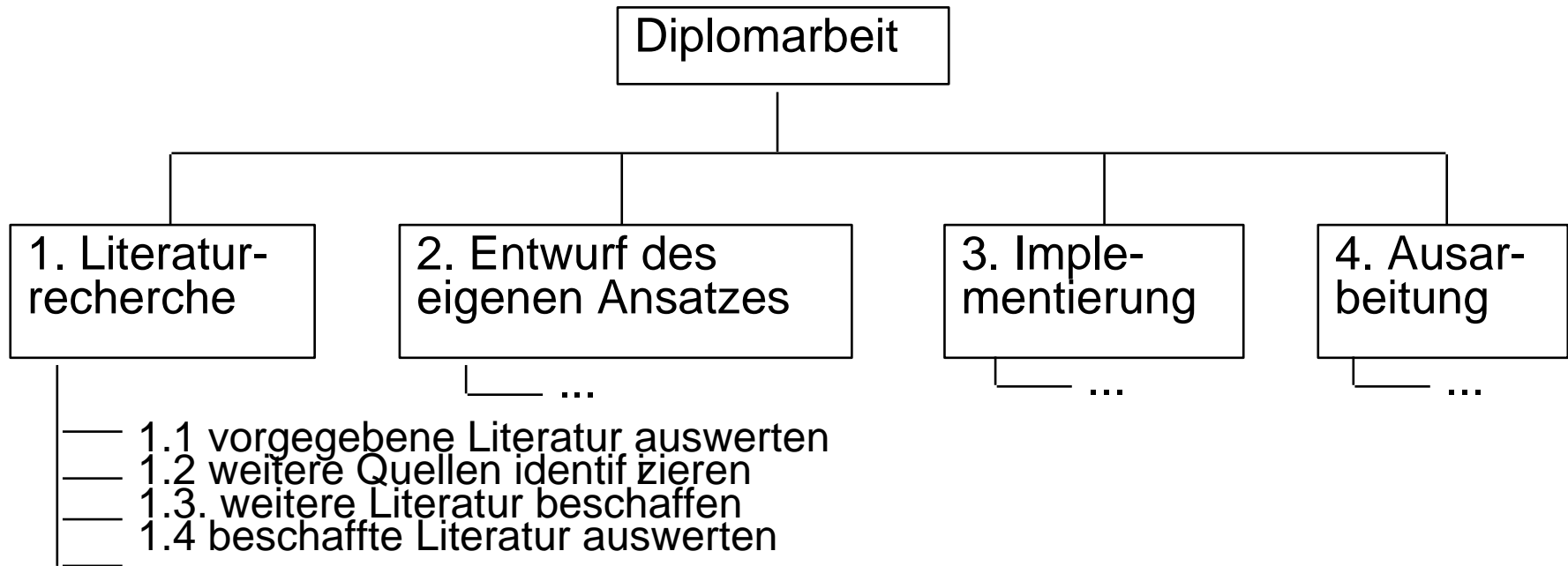
Schweizer Richter und Staatsrechtler, Buchautor und christl. Staatsrechts-Philosoph

Seine Bücher beeinflussten auch K. Adenauer

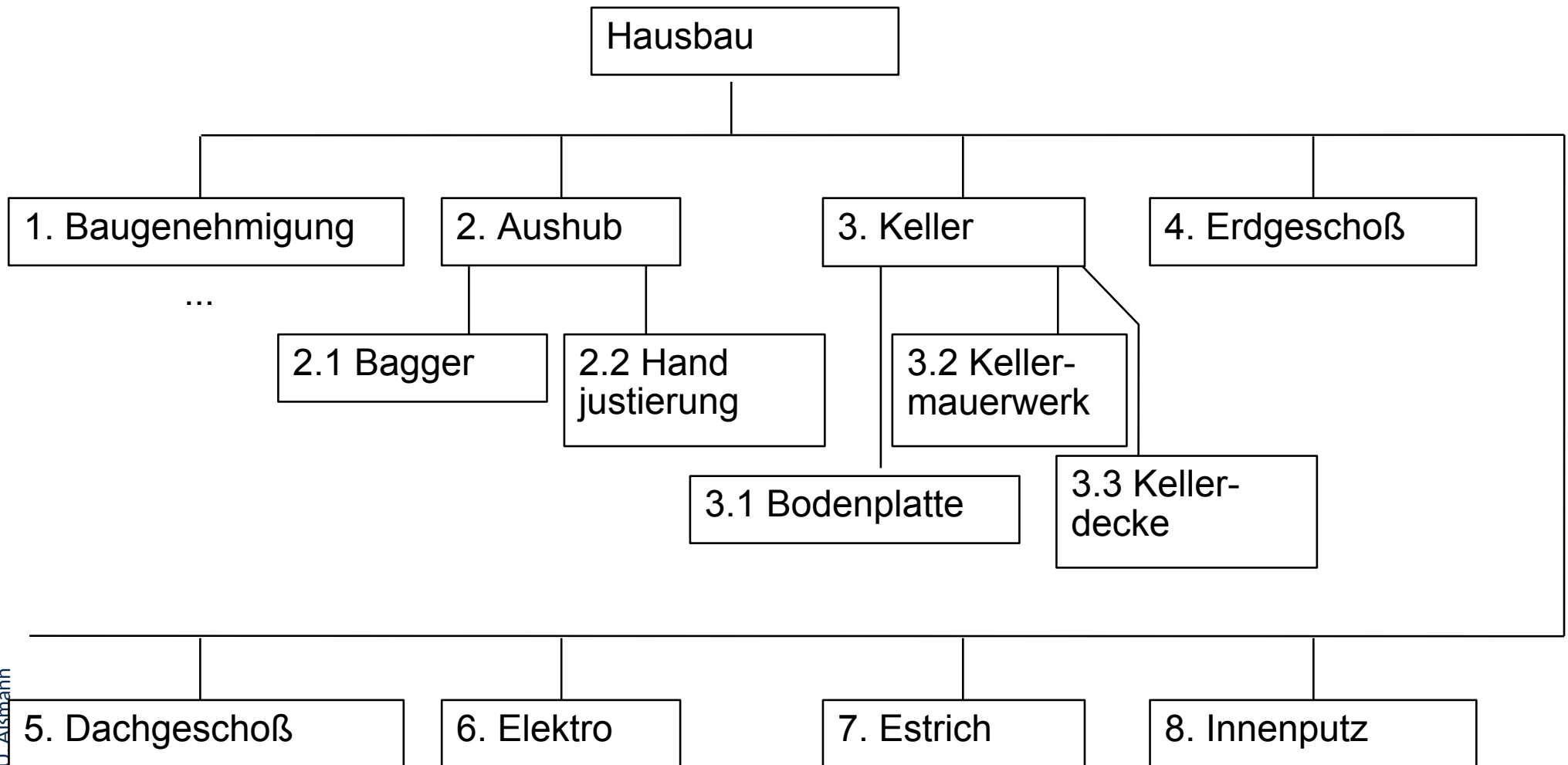


DRESDEN
concept
Exzellenz aus
Wissenschaft
und Kultur

Projektstruktur (“Work Breakdown Structure”): Beispiele



Hausbau

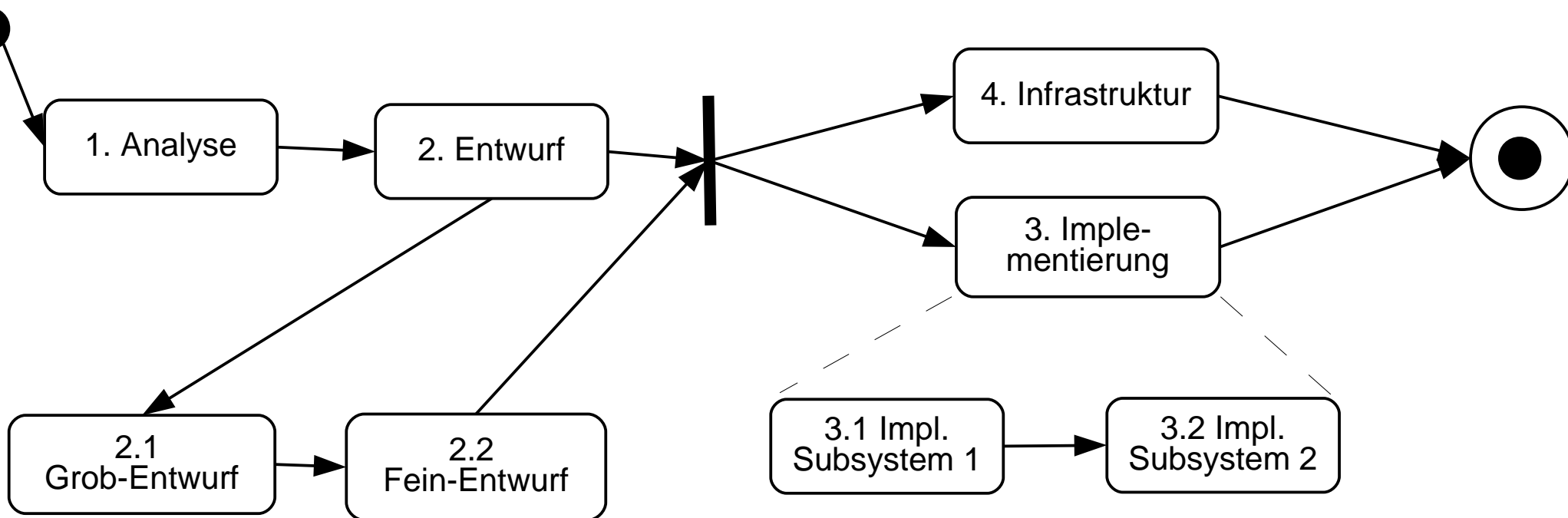


Aufwandsschätzung

- ▶ Schätzungen für:
 - relativen Aufwand der Teilaufgaben
 - absoluten Aufwand für Subsysteme
- ▶ Faustregeln, Erfahrungswerte
- ▶ Techniken der Aufwandsschätzung:
 - Befragung von Entwicklern
 - Klassifikation z.B. durch "Function Point"-Methode
 - Wie viele Teilfunktionen?
 - Wie schwierig ist jede Teilfunktion?
 - Metriken für Spezifikationen
 - "Kalibrierung" durch eigene Erfahrungswerte
- ▶ Mehr in Vorlesung „Softwaremanagement“, SS

Abhängigkeiten

- ▶ Welche Aktivitäten hängen von Ergebnissen anderer Aktivitäten ab? (Abhängigkeitsgraph)
- ▶ Aufwandsschätzung + feste Termine + Abhängigkeiten:
 - Netzplantechniken (z.B. PERT)
 - GANTT-Diagramm
- ▶ Beispiel für Abhängigkeiten, erfaßbar in Aktivitätendiagramm:

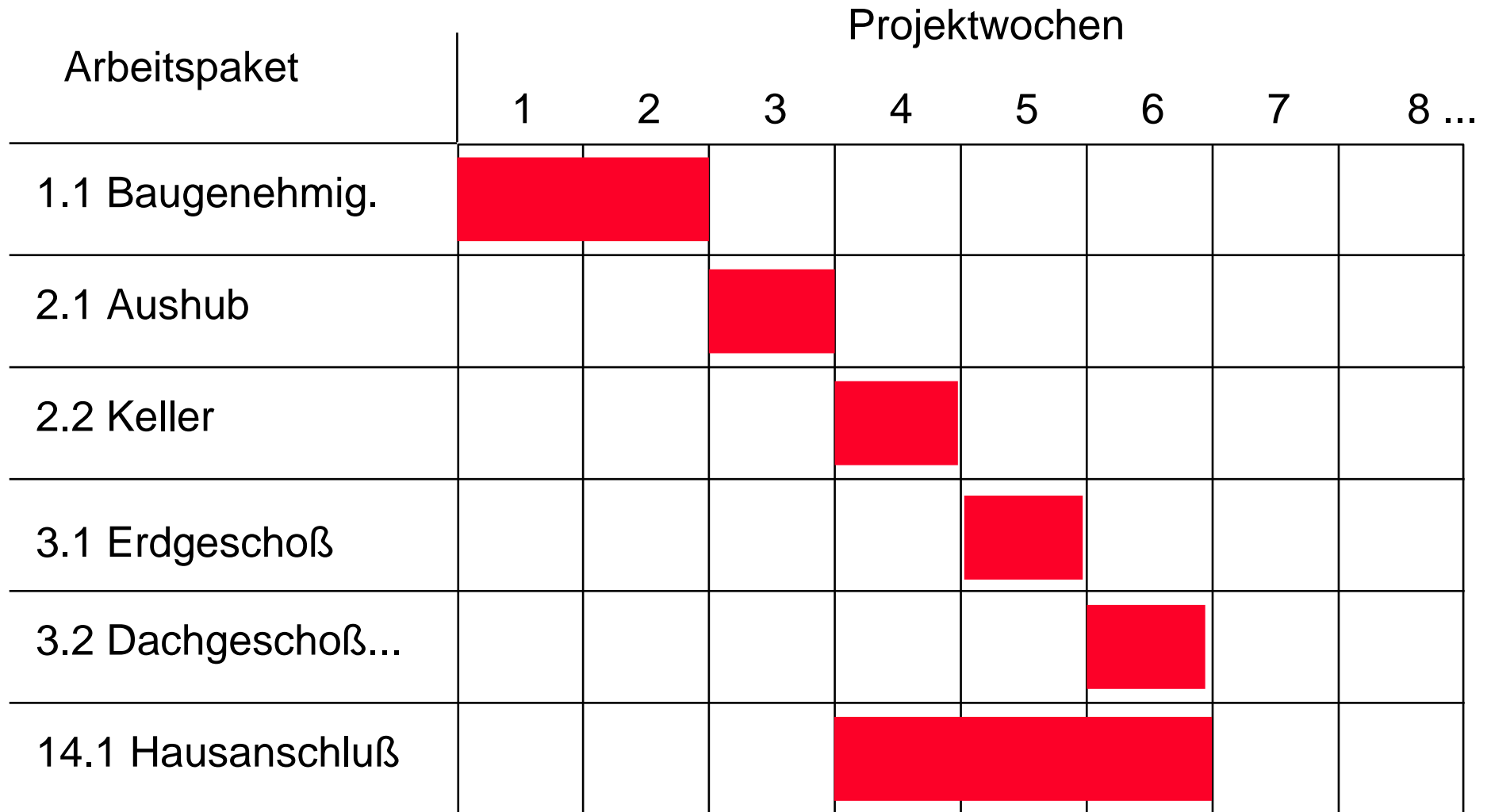


Zeitplanung: Gantt-Diagramm, eine Aktivitätentabelle

Arbeitspaket	Projektwochen							
	1	2	3	4	5	6	7	8 ...
1.1 Analyse	■							
2.1 Grobentwurf			■					
2.2 Feinentwurf				■				
3.1 Impl. Subsys. 1						■		
3.2 ff ...						■		
4.1 Werkzeuge		■						

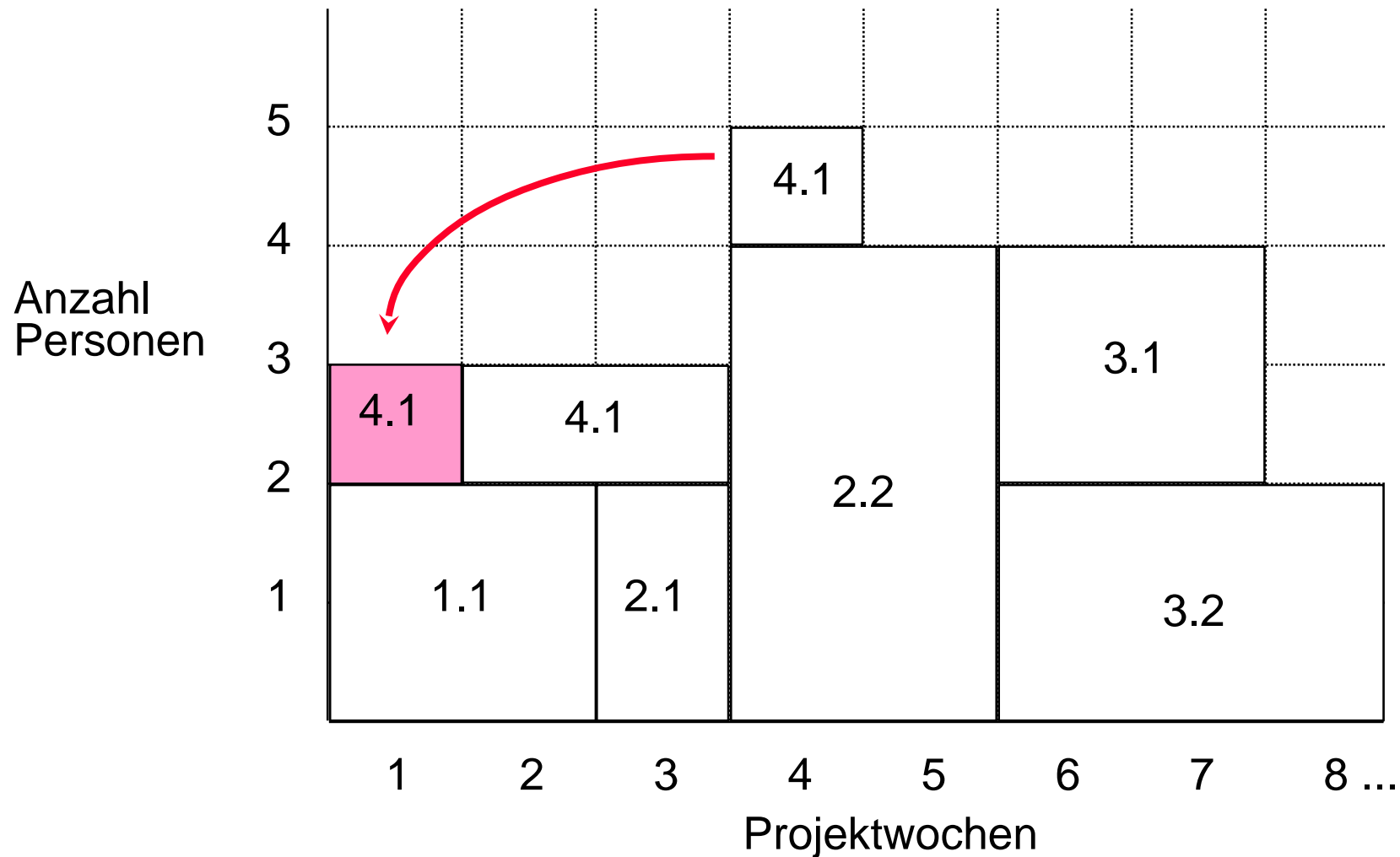
Identifikation *kritischer* und *unkritischer* (4.1, 3.1) Arbeitspakete
(kritisch = Verlängerung verlängert Gesamtprojektdauer)

Zeitplanung Hausbau: Gantt-Diagramm



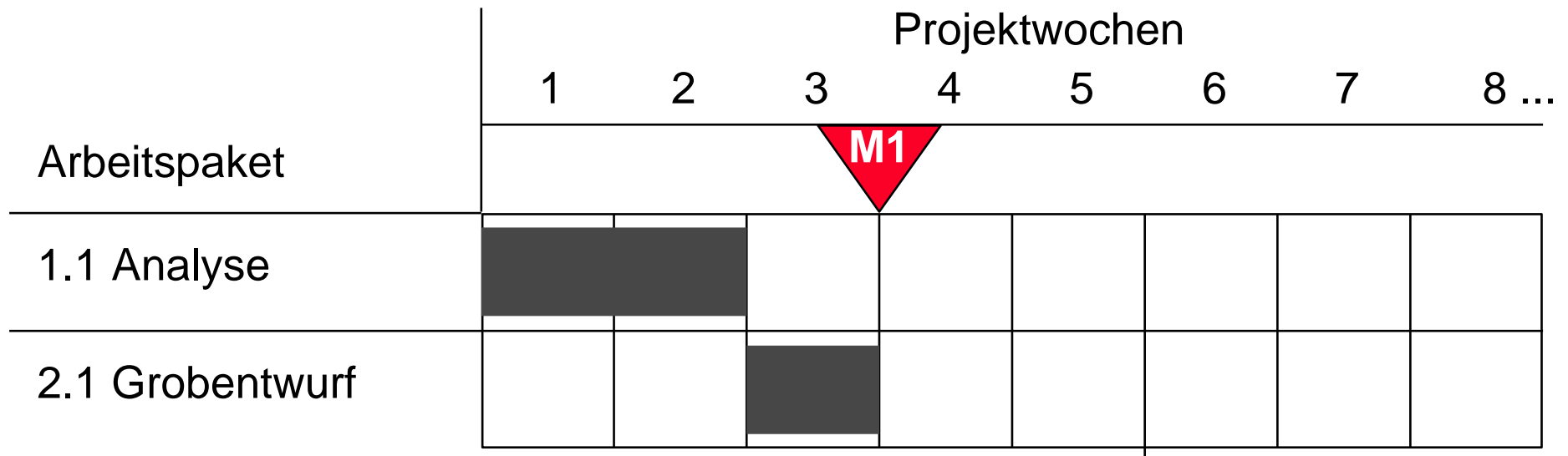
Ressourcenplanung

- ▶ Umplanung mit dem Ziel: Anpassung an vorhandene Ressourcen
- ▶ *Packen* in Flächen über Anz. Personen und Projektwochen



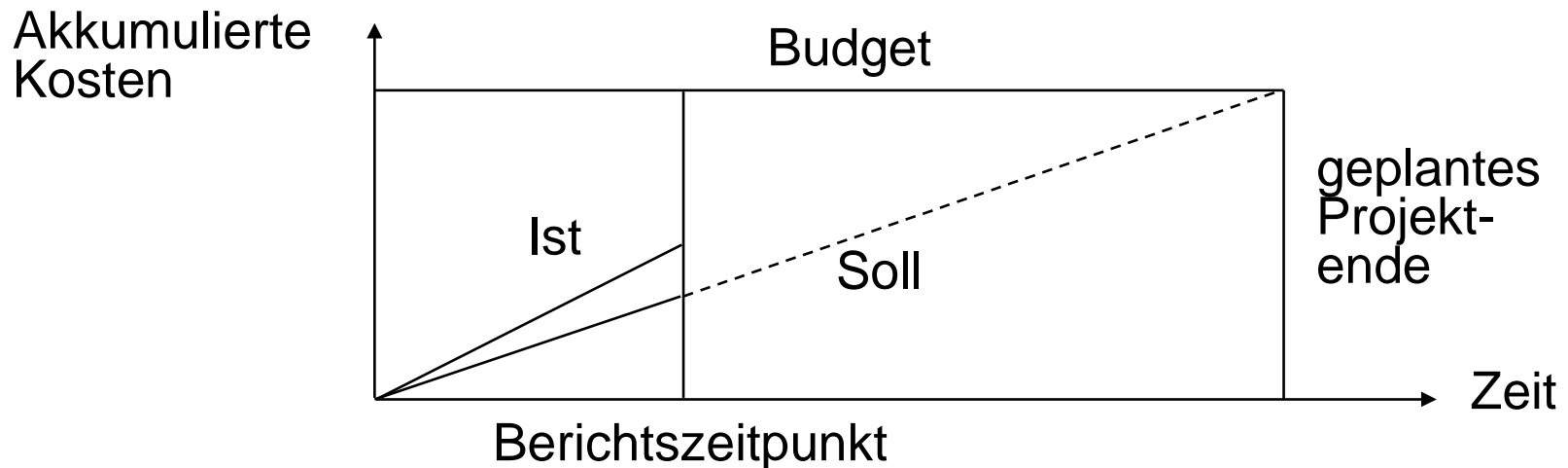
Meilensteine

- ▶ Ein *Meilenstein* ist ein klar definiertes Zwischenresultat, an Hand dessen der Projektfortschritt beurteilt werden kann.
- ▶ Beispiele:
 - "Anforderungsspezifikation zusammen mit Auftraggeber verabschiedet"
 - "Erster Prototyp lauffähig"
 - Schlechtes Beispiel: "Code zu 50% fertig"
- ▶ Meilensteine im Gantt-Diagramm:



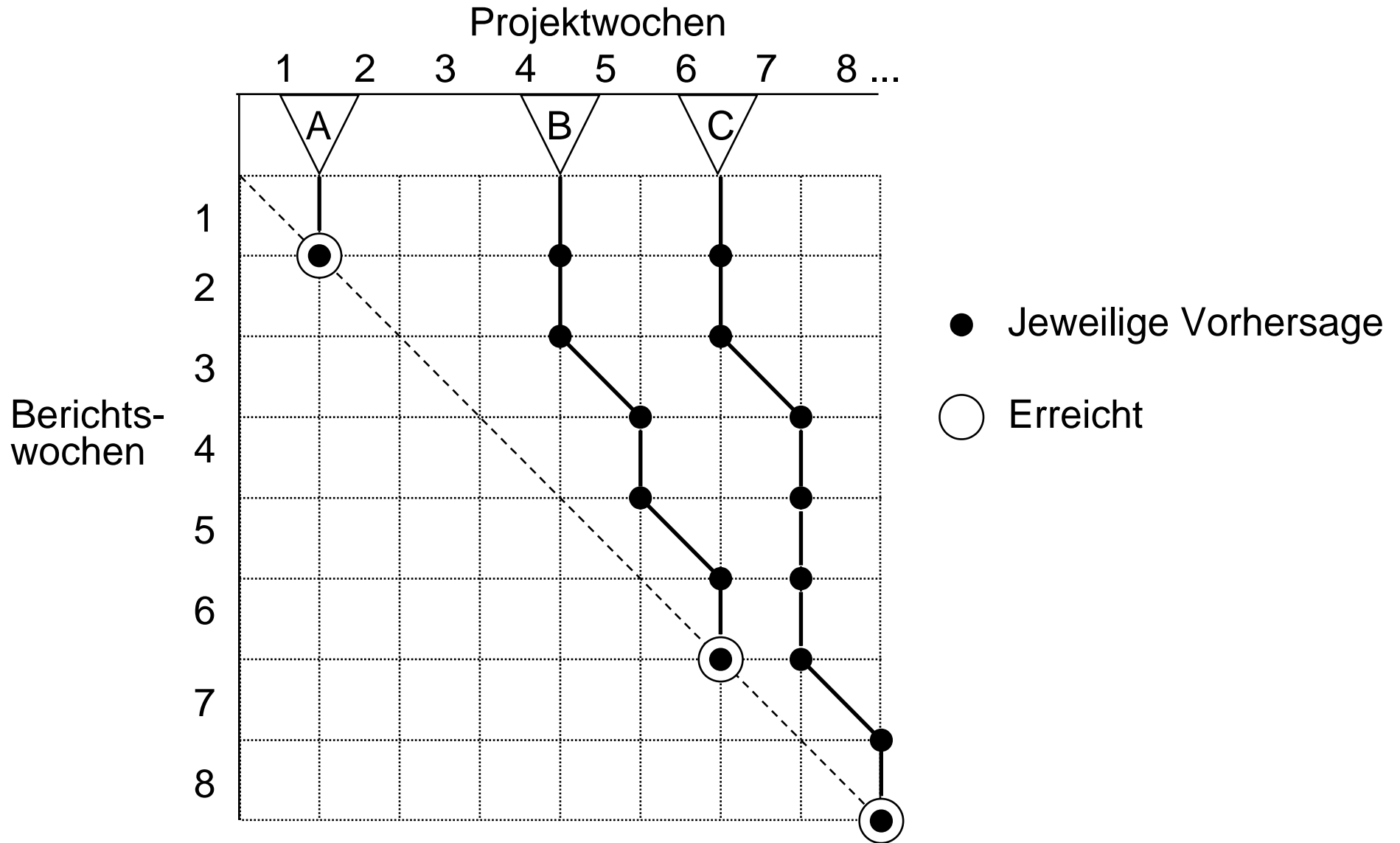
Projektverfolgung

- ▶ Das Projektmanagement muß ein "Frühwarnsystem" für eventuelle Probleme betreiben (Projektverfolgung).
- ▶ Informationsquellen:
 - Laufende (z.B. wöchentliche) Management-Berichte
 - Arbeitszeit-Kontierung
 - Resultate (*deliverables*)
- ▶ Rückkopplung zum Projektteam
 - Regelmäßige Projektbesprechungen
 - Beispiel: Akkumulierter Ressourcenverbrauch



Meilenstein-Trendanalyse

- ▶ Anhand jedes Managementberichts sagt das Management die Meilensteine neu voraus

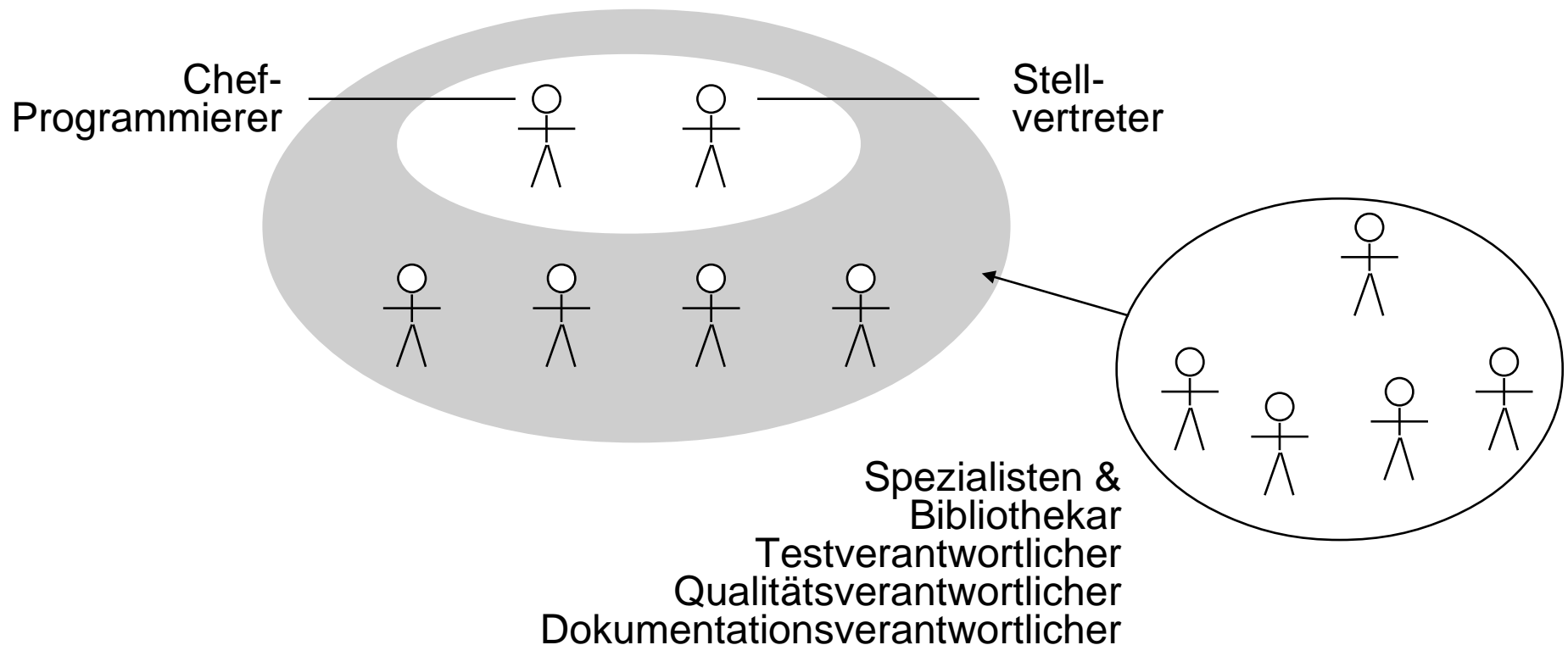


Teamzusammenstellung (Staffing)

14

Softwaretechnologie (ST)

- ▶ Regeln für Teamproduktivität:
 - Optimale Teamgröße: ca. 5-7 Personen
 - Gemischte Qualifikationen
 - Team von externer Kommunikation entlastet
 - Große Projekte aus vielen Teams zusammengesetzt
- ▶ Harlan Mills / Baker 1972: *Chefprogrammierer-Struktur*



Organisation von Sitzungen

- ▶ Vor Sitzungen sollte man immer folgendes (schriftlich) fixieren:
- ▶ **Ziele**
 - Zweck des Treffens (was wollen wir erreichen?)
 - Erfolgskriterien des Treffens (wie können wir kontrollieren, dass wir das Ziel erreicht haben?)
- ▶ **Agenda**
 - Welche Teilnehmer? Haben diese versteckte Zielkonflikte?
 - Zeitplanung: Wie lange welcher Punkt?
- ▶ **Verantwortlicher für ein Ergebnisprotokoll**

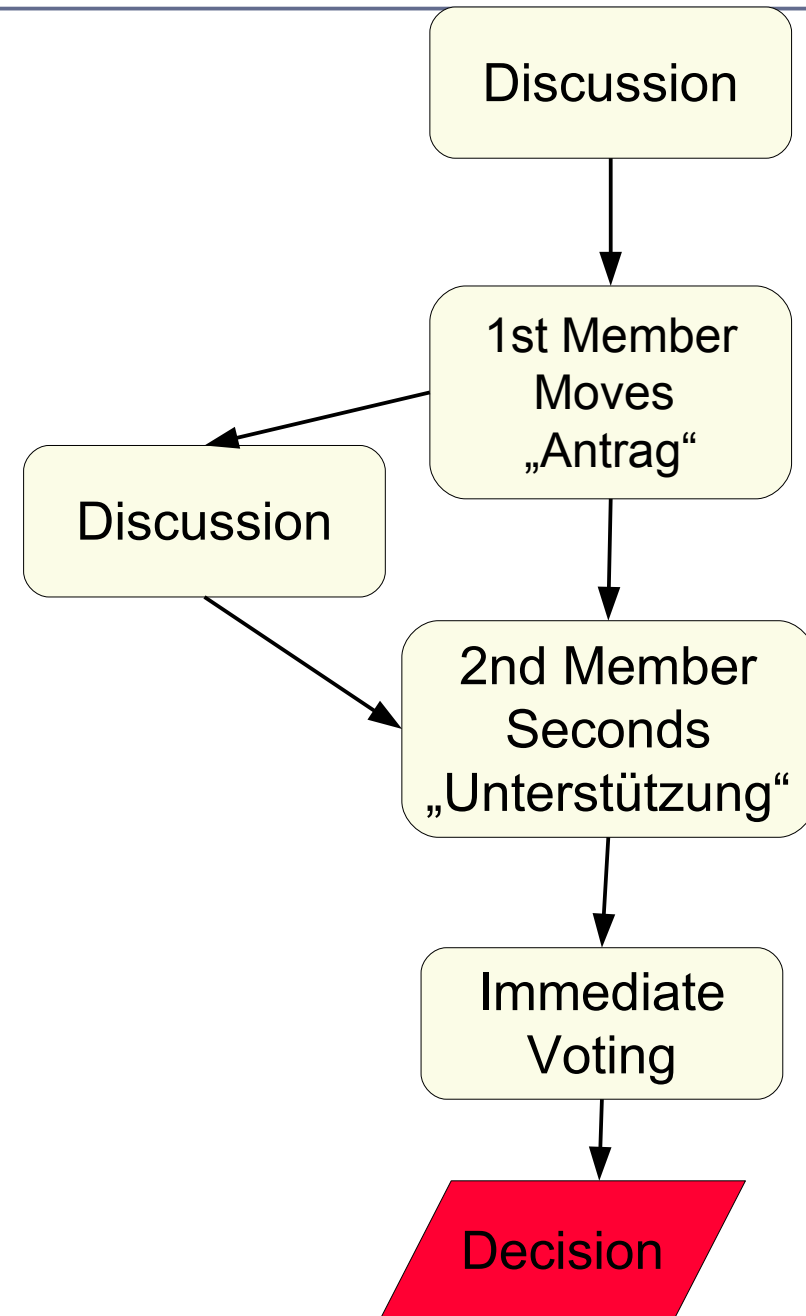
Wie kommt man zu Entschlüssen?

Robert's Rules of Order

16

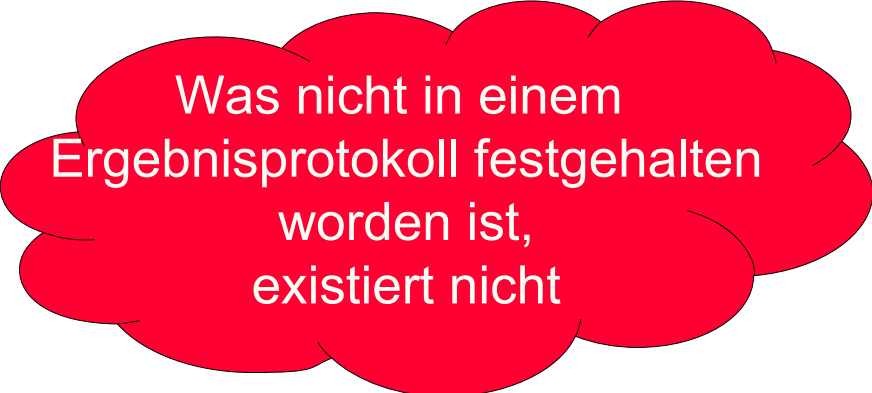
Softwaretechnologie (ST)

- ▶ Demokratische Sitzungen sollten nach “Robert's Rules of Order for Debate” abgehalten werden
 - In USA als “parliamentary procedures” eingeführt, um ineffektive Sitzungen zu vermeiden
 - Jeder Amerikaner kennt sie, denn man lernt sie in der Schule...
- ▶ Definierte Schritte in der Sitzung.
 - “**Movement**” (Antrag zur Abstimmung) wird eingeleitet mit “I move for”
 - “**Secondment**” Antrag muss von zweitem Teilnehmer bestätigt werden “I second”
 - **Voting** Dann muss sofort über den Antrag abgestimmt werden. Wenn niemand unterstützt, entfällt der Abstimmungsantrag.
- ▶ Daneben gibt es etwa 10 weitere Regeln:
 - C. Alan Jennings: Robert's Rules for Dummies (For Dummies (Lifestyles Paperback))



Typische Gliederung eines Ergebnisprotokolls

- ▶ Name der Sitzung
- ▶ Teilnehmer, Moderator, Ort, Zeit
- ▶ Tagesordnung
 - Standard-Tagesordnungspunkte:
 - Protokollkontrolle
 - Bericht über den erreichten Stand
 - Einzelaufgaben
 - Nächster Termin
- ▶ Ergebnisse
 - gegliedert nach Tagesordnungspunkten (TOPs)
 - Beschlüsse
 - Ziele
 - Einzelaufgaben
 - Allokation von Einzelaufgaben an abarbeitende Personen
 - abgelehnte Anträge
 - vertagte Anträge



Was nicht in einem Ergebnisprotokoll festgehalten worden ist, existiert nicht

Einzelaufgaben (*Action Items, Aktivitäten*)

- ▶ Einzelaufgabe (*action item, action point*) besteht aus:
 - Lfd. Nr., Verantwortliche Person
 - Kurztitel
 - Beschreibung
 - Ursprung (Sitzung, auf der Aufgabe definiert wurde)
 - Termin
 - Status (offen, verlängert, erledigt)
- ▶ Liste der Einzelaufgaben wird bei **jedem** Treffen durchgegangen und aktualisiert:
 - Welche Aufgaben sind fällig?
 - Was ist das Ergebnis?
 - Was ist weiter zu tun?
 - Termin verlängern
 - Neue Aufgaben definieren
- Können in einem *issue tracker* verwaltet werden (z.B. Mantis.org)

Aufgabenmanagement-Werkzeuge

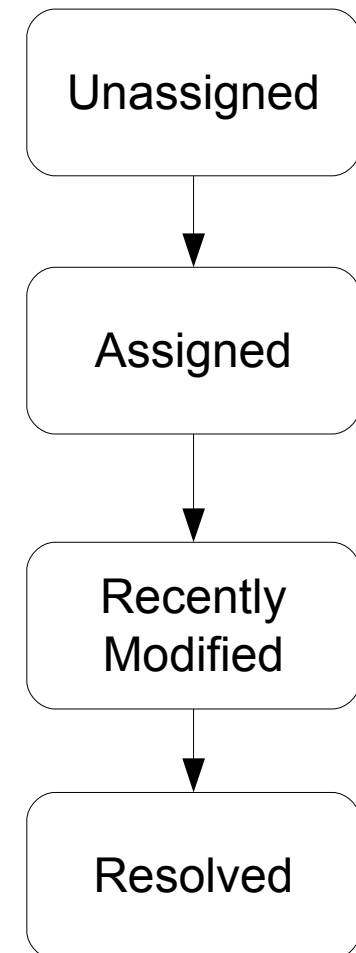
- ▶ **Einzelaufgaben (Aktivitäten)** werden in ein AufgabenmanagementäSystem eingetragen (“ticket system”, “issue management system”)
- ▶ Lesen Sie sich in den Semesterferien in Mantis ein!

Werkzeug	Status	Webadresse
Bugzilla	Mozilla (OSS)	www.bugzilla.org
Mantis	OSS	http://www.mantisbt.org/
JIRA	Atlassian	http://atlassian.com/software/jira
codeBeamer	IntLand Software	http://intland.com/products/codebeamer/overview/
RedMine	OSS	http://en.wikipedia.org/wiki/Redmine
Team Foundation Server	Microsoft	http://en.wikipedia.org/wiki/Team_Foundation_Server

http://en.wikipedia.org/wiki/Comparison_of_issue_tracking_systems

Aufgabenmanagement (Fehler und Änderungen) mit Mantis

- ▶ Mantis ist ein webbasiertes Aufgabenmanagement-System (issue tracking system)
 - Zustandsmaschine für Fehler und Änderungswünsche
 - Generierung von emails über Statusänderungen
 - Visualisierung von Zuständen und Bearbeitern



Überblick über Aufgaben [www.mantisbt.org]

Viewing Issues (1 - 50 / 1520) [Print Reports] [CSV Export]

P	ID	US\$	#	Category	Severity	Status	Updated	Summary
	0005069		3	GUI	minor	resolved (MacGyver)	2009-01-28	Invalid Password
	0005095			Other	minor	assigned (mow)	2009-01-28	error prueba
	0005091			Other	trivial	confirmed (ossgevalt)	2009-01-28	foo bar baz
	0004707			Other	minor	resolved (000willberty)	2009-01-28	relationships I
	0005093			GUI	minor	assigned (enzyme)	2009-01-28	Test fichier attache
	0005082			Other	minor	assigned (121212)	2009-01-28	test
	0005094			GUI	minor	new	2009-01-28	Test fichier attache
	0005092			GUI	minor	new	2009-01-28	Test fichier attache
	0005090			GUI	minor	assigned (flandersen)	2009-01-28	Needs attention
	0005089		3	Website	major	resolved (Russell)	2009-01-28	Problems loading JavaScript on Main.html
	0005076		1	GUI	feature	assigned (darksaboteur)	2009-01-28	I can't poo...
	0005088			GUI	tweak	assigned (abarbosa)	2009-01-27	sadasdasda
	0005087			GUI	trivial	assigned (ramyap)	2009-01-27	asdasddsad
	0005086		9	GUI	trivial	assigned (Element)	2009-01-27	Detailbereich wird nicht angezeigt
	0005083		1	Other	minor	acknowledged (aaloc)	2009-01-27	test
	0005085			GUI	major	assigned (celso)	2009-01-27	teste com erro e associaçao com arquivo de repositório
	0005081			GUI	minor	resolved (deepak84)	2009-01-27	Sql Error
	0005080			GUI	feature	assigned (mmiot)	2009-01-27	????
	0005079			GUI	minor	assigned (kedar)	2009-01-26	test
	0005078		4	GUI	text	assigned (jodji)	2009-01-26	test
	0005077			GUI	minor	new	2009-01-26	cant abc
	0005065		3	GUI	tweak	new	2009-01-26	hhhh
	0005074		2	Other	minor	assigned (maxadmin)	2009-01-26	VIZ funktioniert nicht
	0005075			Website	major	assigned (patriciagomes)	2009-01-26	Erro no site
	0005071		2	Website	major	assigned (deepak84)	2009-01-26	Website Issue
	0005072			GUI	minor	assigned (trm)	2009-01-25	Alles Schiefe

Überblick über Zustände der Aufgaben

The screenshot displays the MantisBT web interface. At the top, there is a navigation bar with the Mantis logo and the text "MANTIS". Below the logo, it says "Anonymous | Login | Signup for a new account" and the date "2009-01-28 14:21 EST". The project is set to "Demo". A search bar contains the text "ebert". The main content area is divided into three sections: "Unassigned", "Resolved", and "Recently Modified". Each section contains a list of tasks with their IDs, descriptions, and dates.

Unassigned [^] (1 - 10 / 271)

0005094	Test fichier attache GUI - 2009-01-28 09:01
0005092	Test fichier attache GUI - 2009-01-28 09:00
0005077	cant abc GUI - 2009-01-26 10:38
0005065	hhhh GUI - 2009-01-26 10:07
0004953	dede GUI - 2009-01-16 00:04
0004921	test da Bologna GUI - 2009-01-16 00:03
0004851	dede GUI - 2009-01-16 00:03
0004845	Enhance CP/M to support 64 bit processors Other - 2009-01-16 00:03
0004835	sddd GUI - 2009-01-16 00:03
0004832	???? GUI - 2009-01-16 00:03

Resolved [^] (1 - 10 / 154)

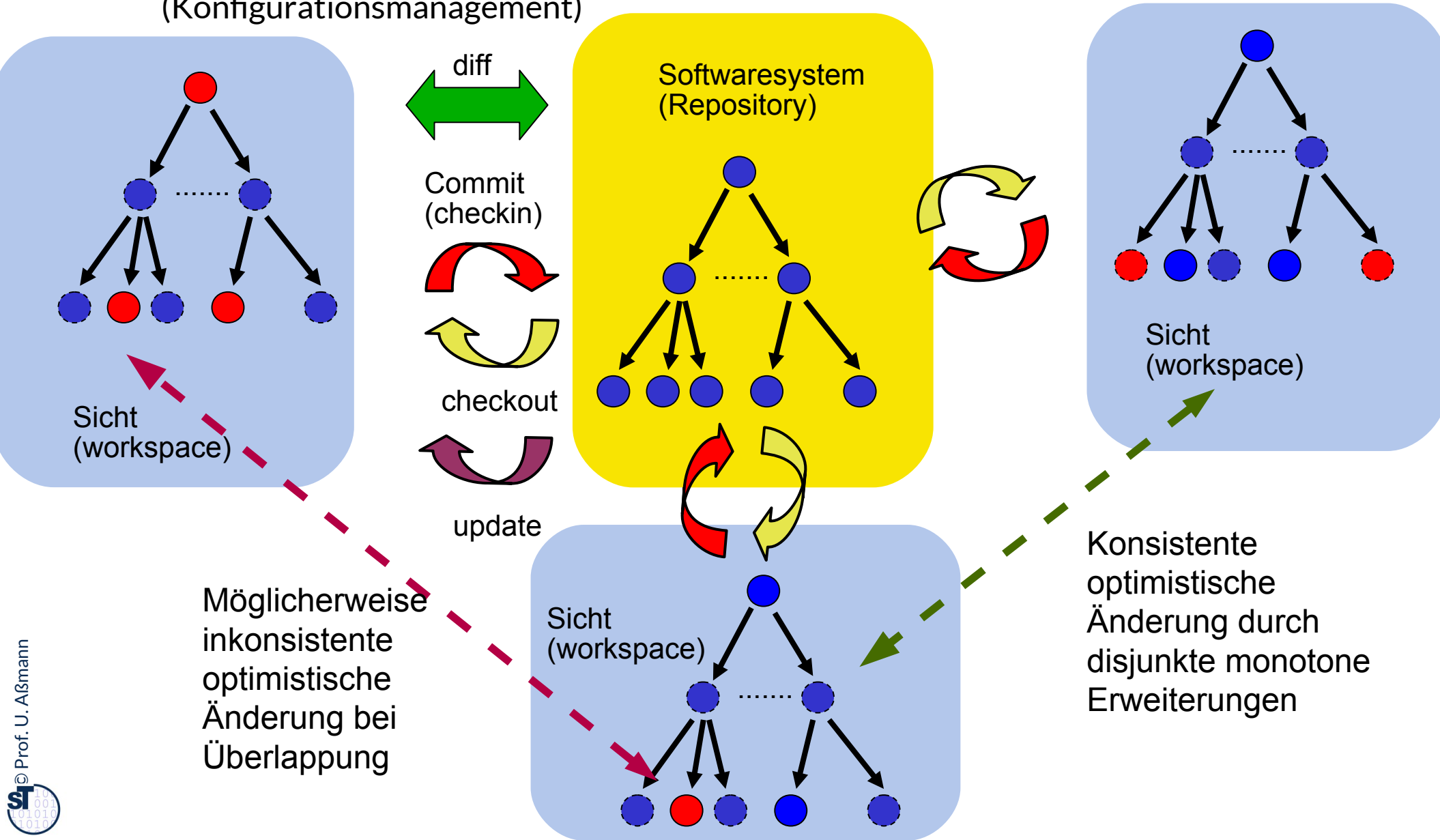
0004707	relationships I Other - 2009-01-28 13:01
0005089	Problems loading JavaScript on Main.html Website - 2009-01-28 04:27
0005081	Sql Error GUI - 2009-01-27 09:53
0004728	test test GUI - 2009-01-22 17:32
0004934	Hello Puur Website - 2009-01-16 00:04
0004915	Hello Puur Website - 2009-01-16 00:03
0004839	Update error Other - 2009-01-16 00:03
0004834	sum GUI - 2009-01-16 00:03
0004556	Testing, checking, Knowing.. Other - 2009-01-16 00:03
0004766	urgent FAIL GUI - 2008-12-23 10:10

Recently Modified [^] (1 - 10 / 2234)

0005069	Invalid Password GUI - 2009-01-28 14:20
0005095	errorr prueba Other - 2009-01-28 13:14
0005091	foo bar baz Other - 2009-01-28 13:04
0004707	relationships I Other - 2009-01-28 13:01
0005096	errorr prueba Website - 2009-01-28 12:57
0005093	Test fichier attache GUI - 2009-01-28 10:22
0005082	test Other - 2009-01-28 09:43
0005094	Test fichier attache GUI - 2009-01-28 09:01
0005092	Test fichier attache GUI - 2009-01-28 09:00
0005090	Needs attention

Zusammenarbeiten mit Konfigurationsmanagement

- ▶ Bei paralleler Bearbeitung müssen Sichten konsistent gehalten werden (Konfigurationsmanagement)



	Werkzeug		URL
Dateibaum- basiert	cvs	OSS	http://www.cvshome.org
	git	OSS	Linus Thorvalds, www.git-scm.com
Datenbank- basiert	ClearCase	IBM/Rational	http://www.rational.com/products
	Visual SourceSafe	Microsoft	http://www.eu.microsoft.com/germany/produkte
beides	subversion	OSS	http://subversion.tigris.org
andere	Synergy	IBM Telelogic	http://www.telelogic.com/product/synergy
	mercurial	Selenic, OSS	http://mercurial.selenic.com/
	InStep	microTOOL GmbH, Berlin	http://www.microTOOL.de

Konkrete Aufgaben zur Vorbereitung des Praktikums

25

Softwaretechnologie (ST)

- ▶ Einlesen:
- ▶ Mantis
- ▶ git oder subversion



50.2 Vorgehensmodelle (Phasenmodelle)



- ▶ Zuser Kap. 1-3 *oder*
- ▶ Ghezzi Chapter 1 *oder*
- ▶ Pfleeger Chapter 1; Chap 8.1

Vorgehensmodell (engl. *process model*)

- Strukturiertes Modell zum Erstellen von Software

Phasenmodell

- Vorgehensmodell, das den Herstellungsprozesses in definierte und abgegrenzte Phasen einteilt
- Vorgabe einer Reihenfolge in der Bearbeitung der Phasen

Wie gehe ich vor, um Software zu entwickeln?

- ▶ Ad hoc

- ▶ Es lief schon oft schief...
 - Denver International Airport, Krise 1993-95
 - Hamburger Güterbahnhof 1995
 - Toll Collect 2004
 - Fiscus 2004

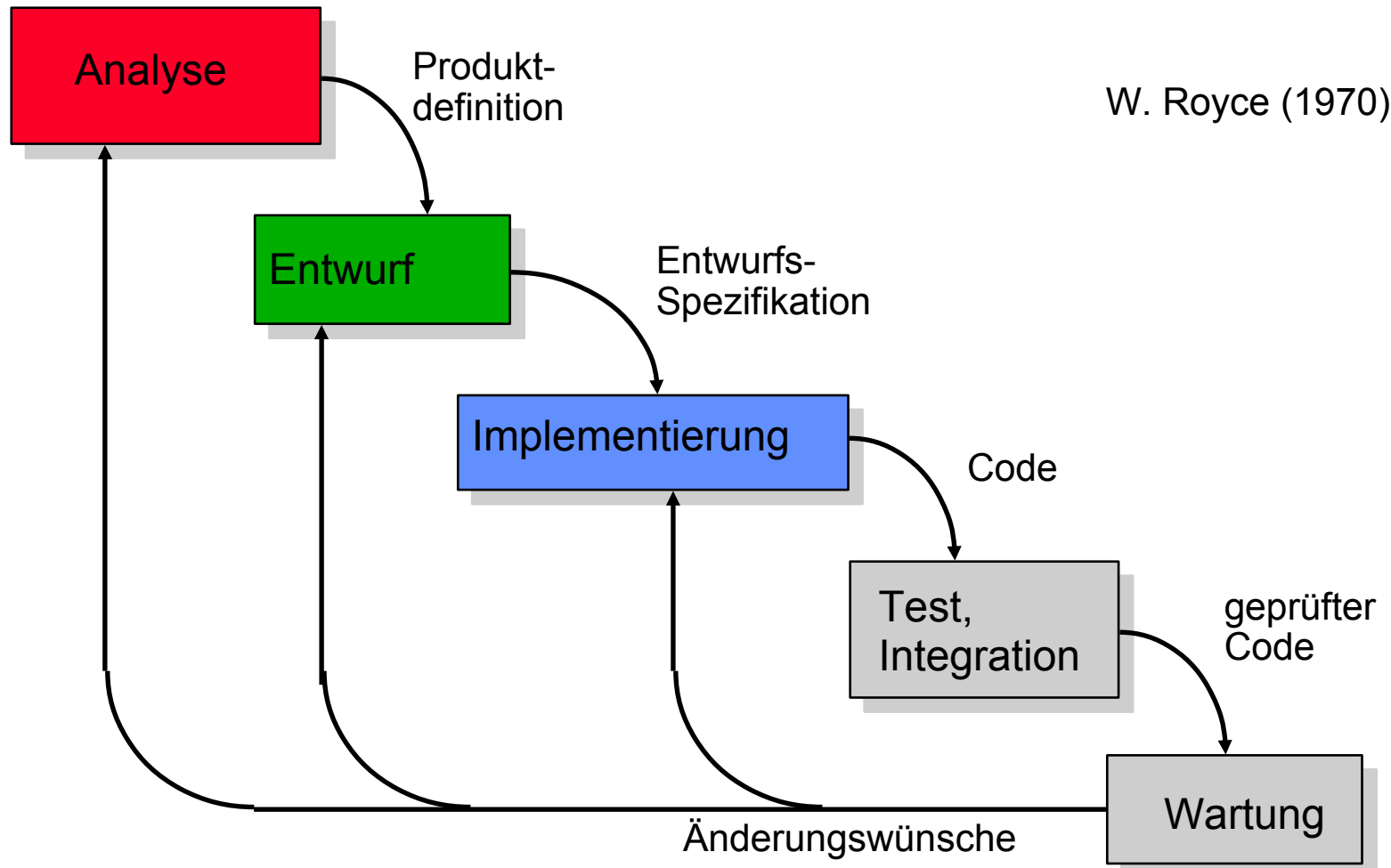
- ▶ Gibt es nicht irgendwelche Hilfen, strukturiert vorzugehen?

Vorgehen nach einem “Phasenmodell”

- ▶ **Phasenmodell** (*process model, software development life cycle*)
 - Einteilung des Herstellungsprozesses für ein (Software-) Produkt in definierte und abgegrenzte Abschnitte, abgegrenzt durch **Meilensteine**
 - Grobgliederung: Phasen (*phases*)
 - Feingliederung: Schritte (*stages, steps*)
 - Vorgabe einer Reihenfolge in der Bearbeitung der Phasen
 - Richtlinie für die Definition von Zwischenergebnissen
 - Detailliertes Phasenmodell + Zwischenergebnisdefinition = „Vorgehensmodell“

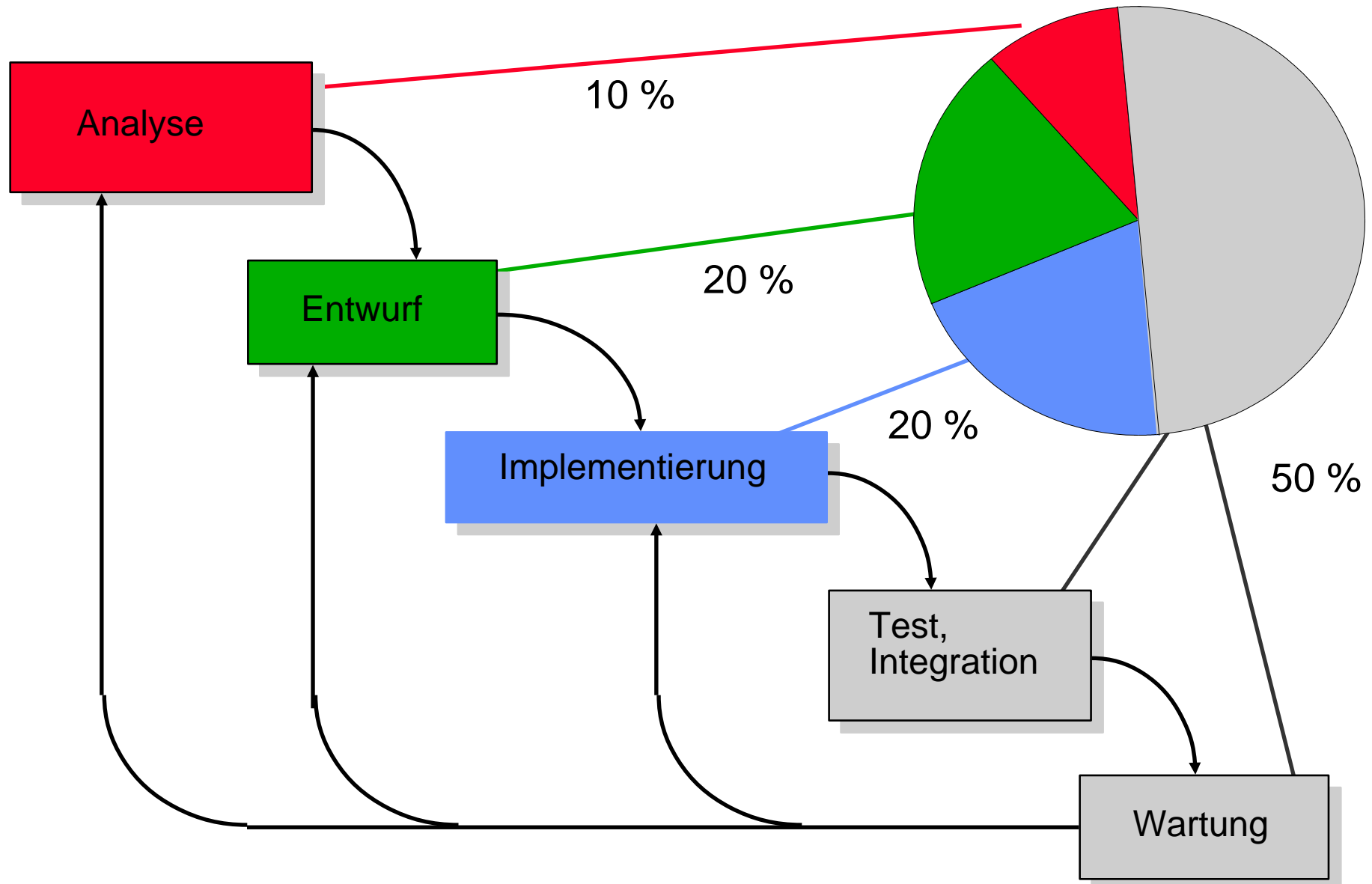
- ▶ Grundaktivitäten:
 - Analyse
 - Entwurf
 - Implementierung
 - Validation (v.a. Test, Integration)
 - Evolution (v.a. Wartung)

Wasserfall-Modell (mit Iterationen)

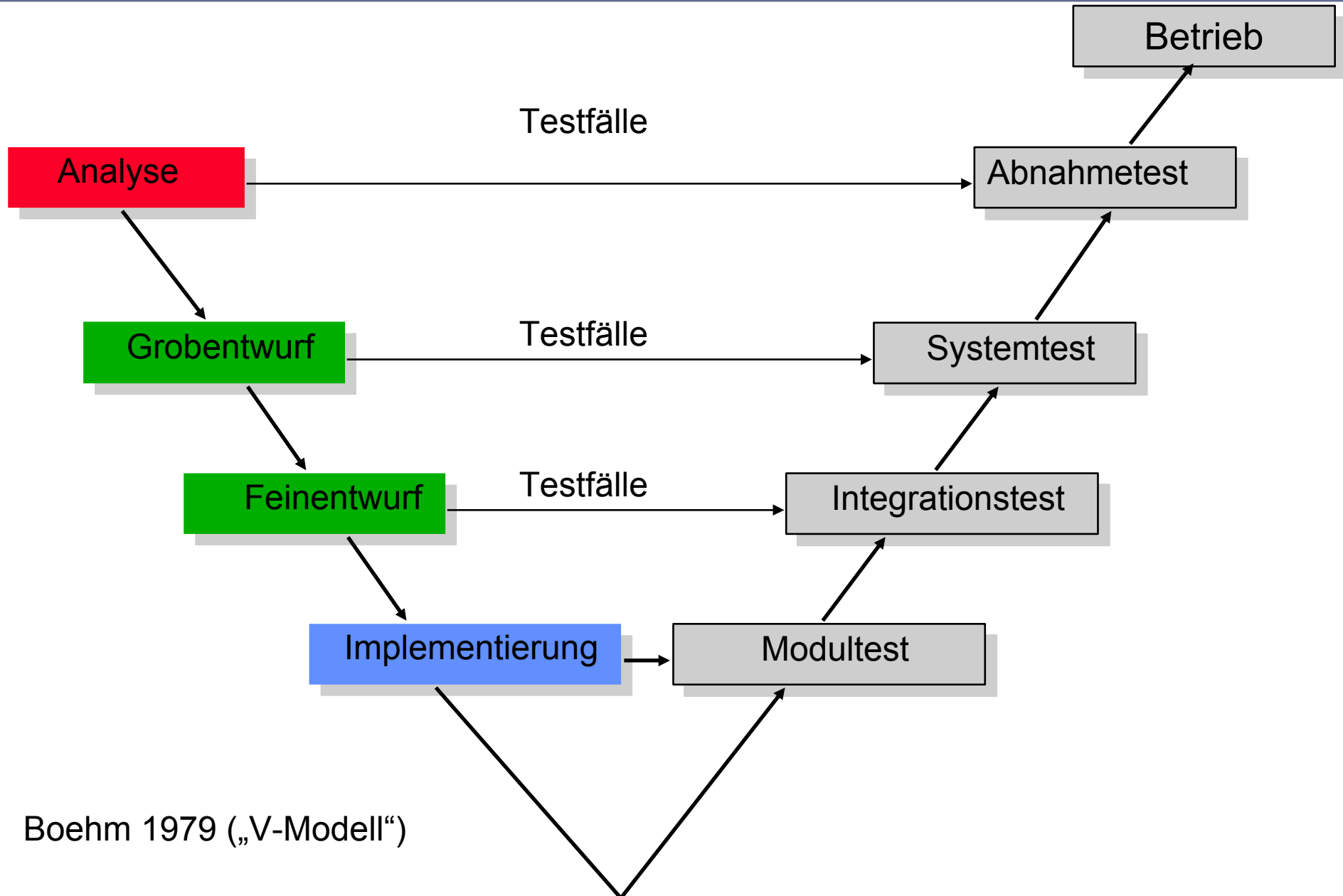


- ▶ Das Wasserfallmodell ist nicht realistisch. Für ein Produkt müssen, schon um des Geschäftsmodell willen, Verbesserungen (Lebenszyklen) eingeplant werden
- ▶ Ein Lebenszyklus dauert i.D. 2 Jahre
- ▶ Dennoch muss ein Softwareingenieur den “Wasserfall” beherrschen, denn viele andere Vorgehen setzen darauf auf

Ungefähre Verteilung des Arbeitsaufwandes

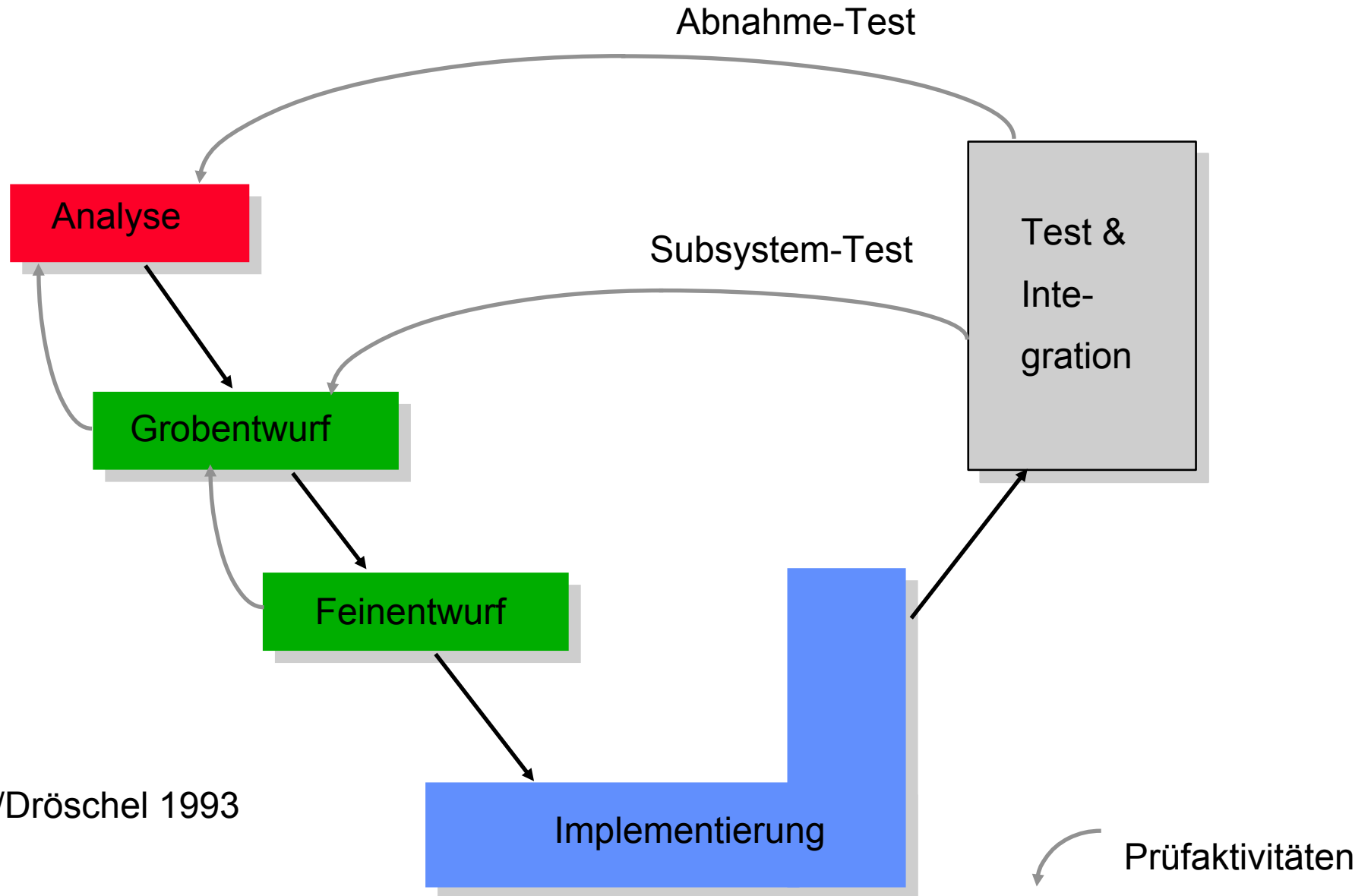


Qualitätssicherung im einfachen V-Modell



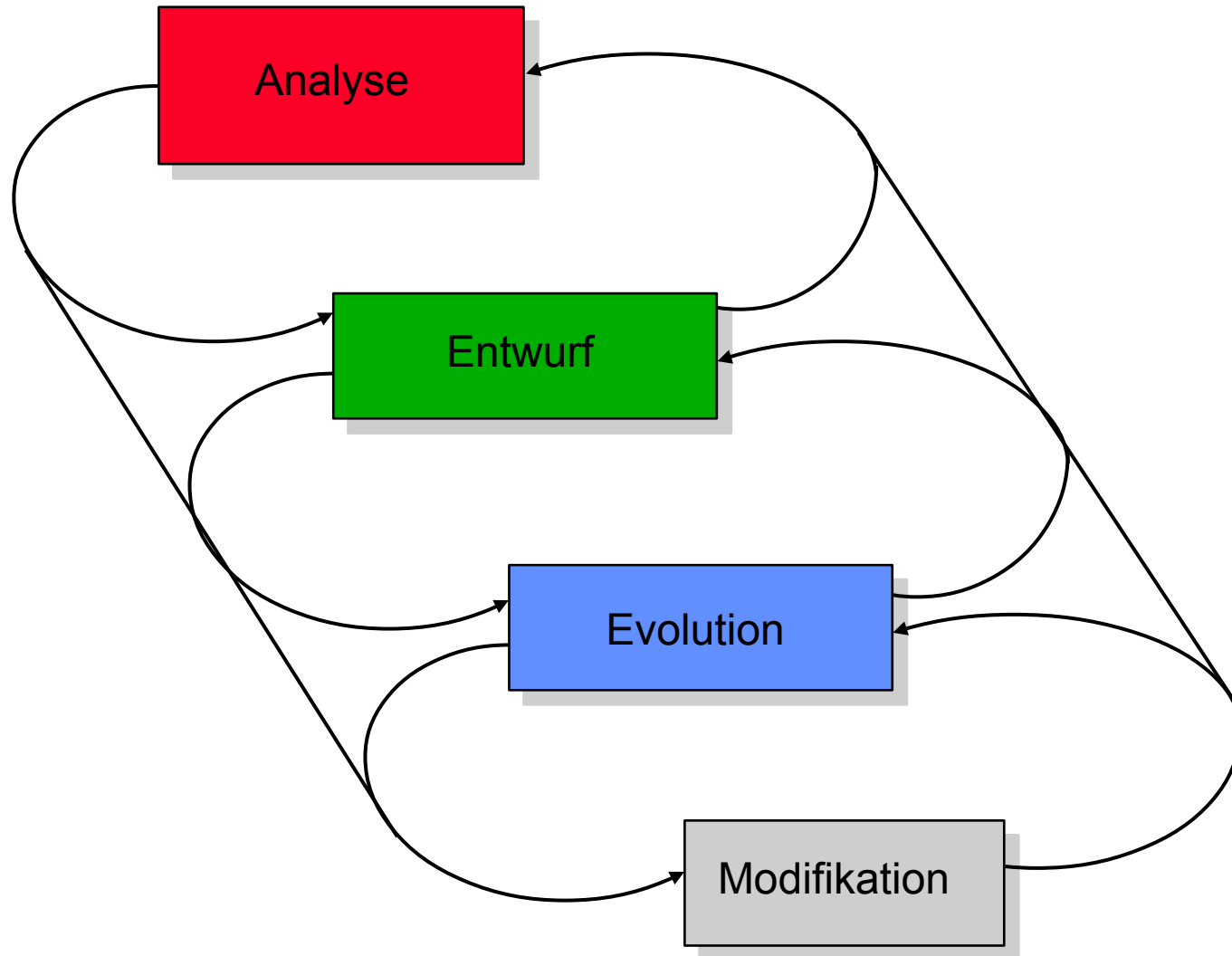
Boehm 1979 („V-Modell“)

V-Modell des BMI (vereinfacht)



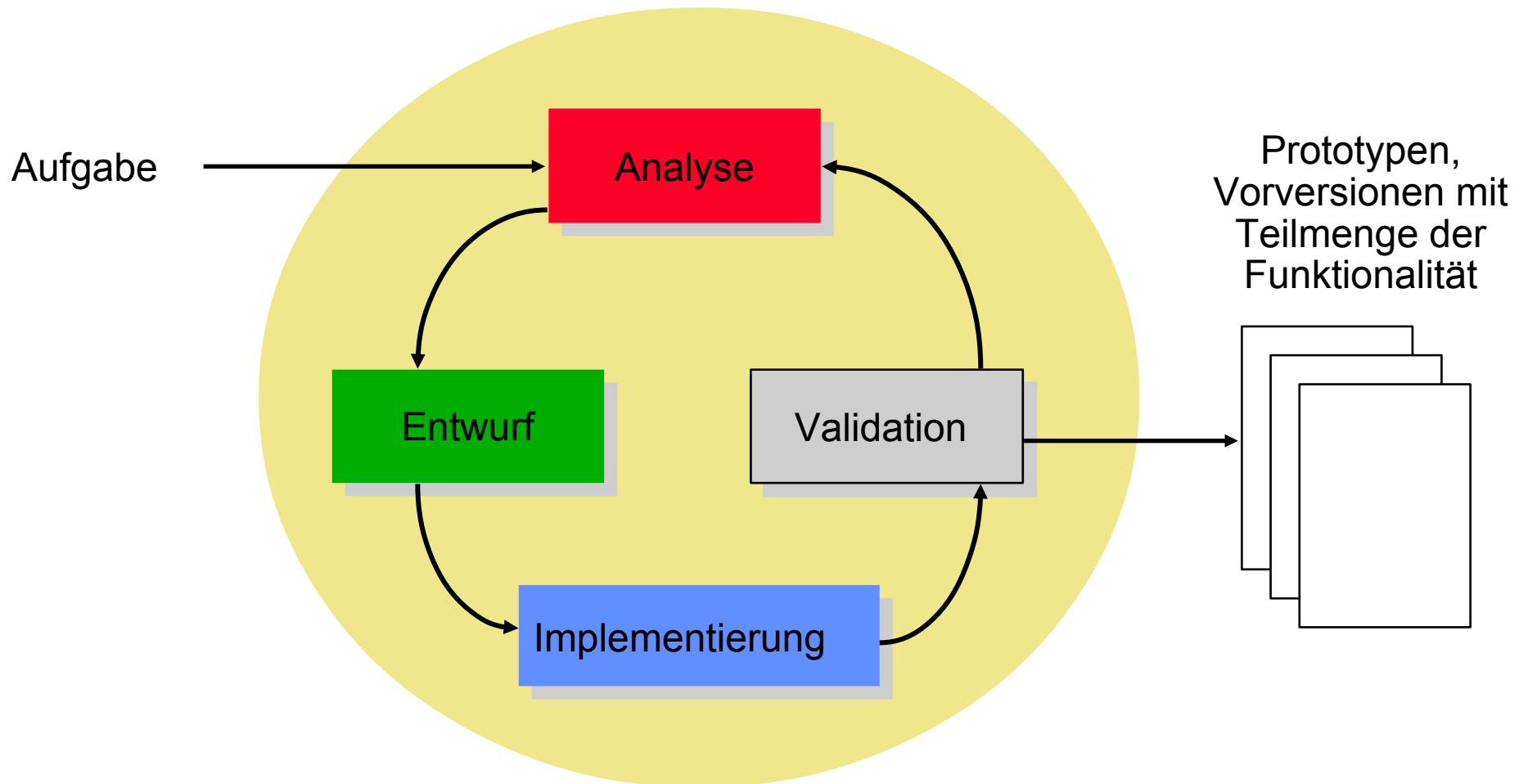
Bröhl/Dröschel 1993

Inkrementelle (evolutionäre, agile) Entwicklung



Evolutionäre Entwicklung

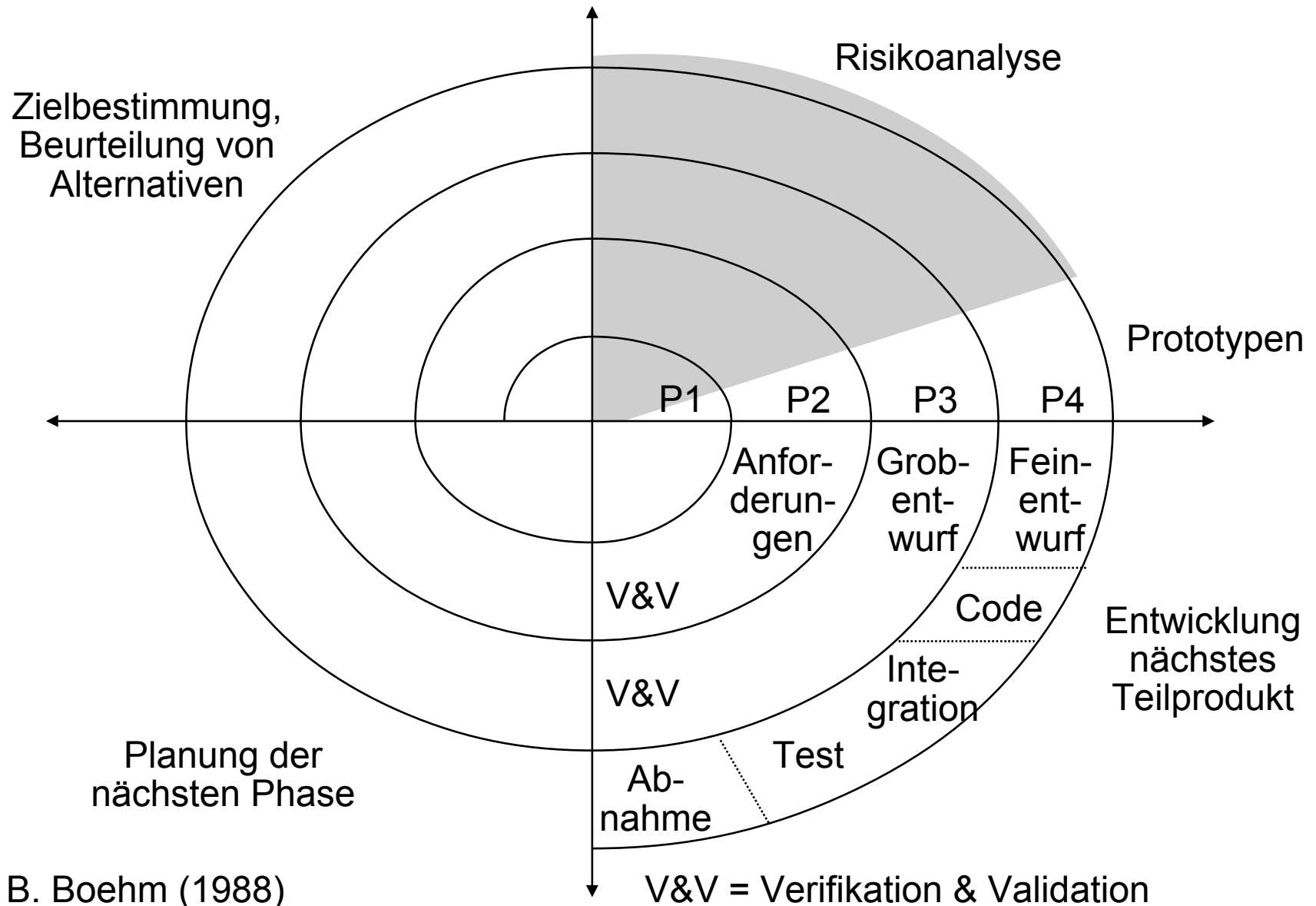
- ▶ Typisch für kleinere Projekte oder experimentelle Systeme
- ▶ Bei Objektorientierung auch für größere Projekte anwendbar ?



eXtreme Programming (XP)

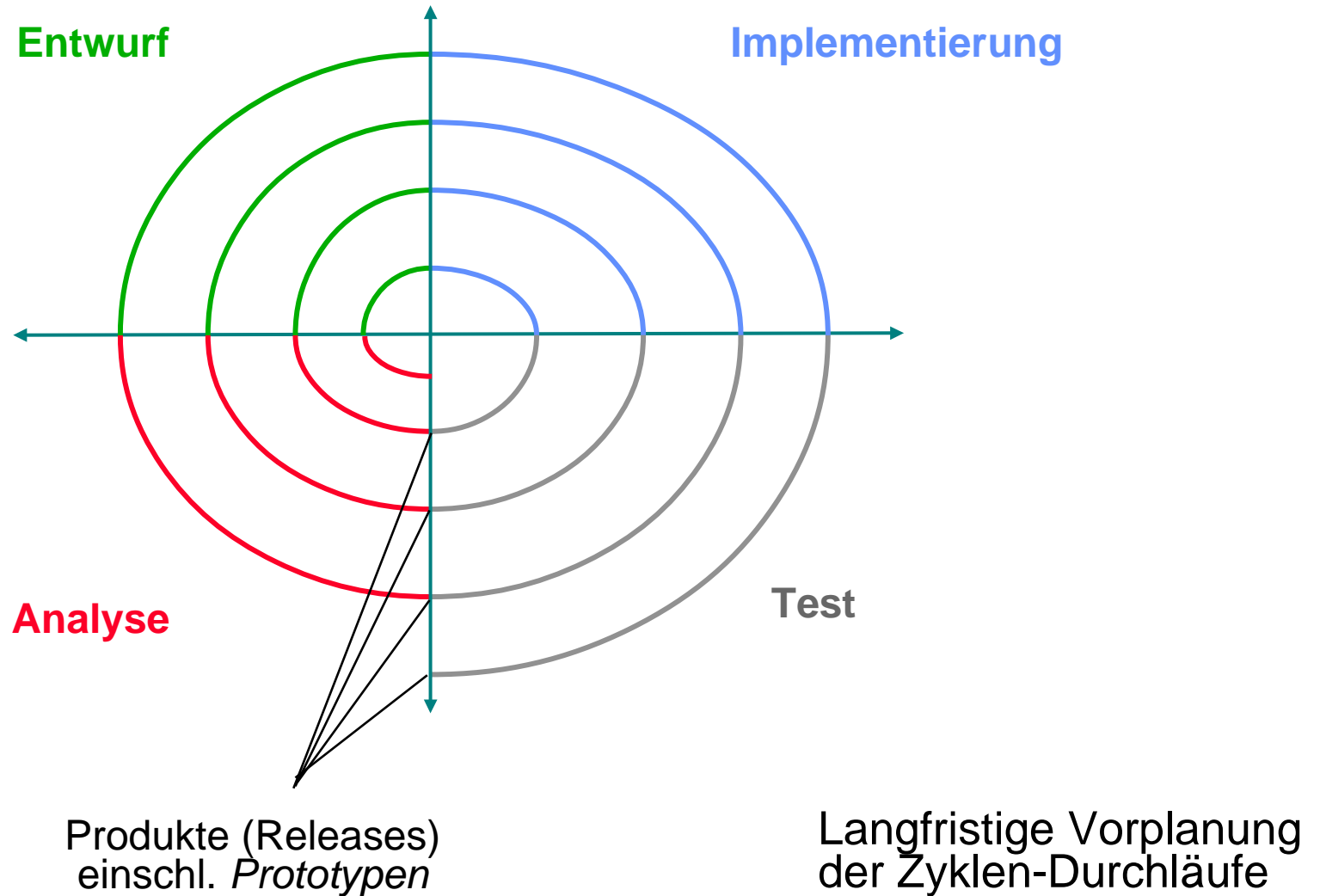
- ▶ Kontrovers diskutiertete Entwicklungsmethodik (Kent Beck)
 - Konsequente evolutionäre Entwicklung
 - Der Programmcode ist das Analyseergebnis, das Entwurfsdokument und die Dokumentation. Code wird permanent (Tagesrhythmus) lauffähig gehalten
 - Diszipliniertes und automatisiertes Testen als Qualitätssicherung
 - Diverse weitere innovative Techniken (z.B. Paar-Programmierung)
 - liefert schnell Ergebnisse, aber u.U. auf Kosten der Langlebigkeit
 - kann prinzipiell mit traditionelleren Analyse- und Entwurfstechniken kombiniert werden
- ▶ Nachteile
 - wird manchmal als Gegenbewegung zu sauberem Softwareentwurf **miß**verstanden
 - ist nur geeignet für relativ überschaubare, isolierte Anwendungen
- ▶ "Agile" Softwareentwicklung (www.agilemanifesto.org):
 - weitere Ansätze, z.B. Crystal, Scrum

Spiralmodell



B. Boehm (1988)

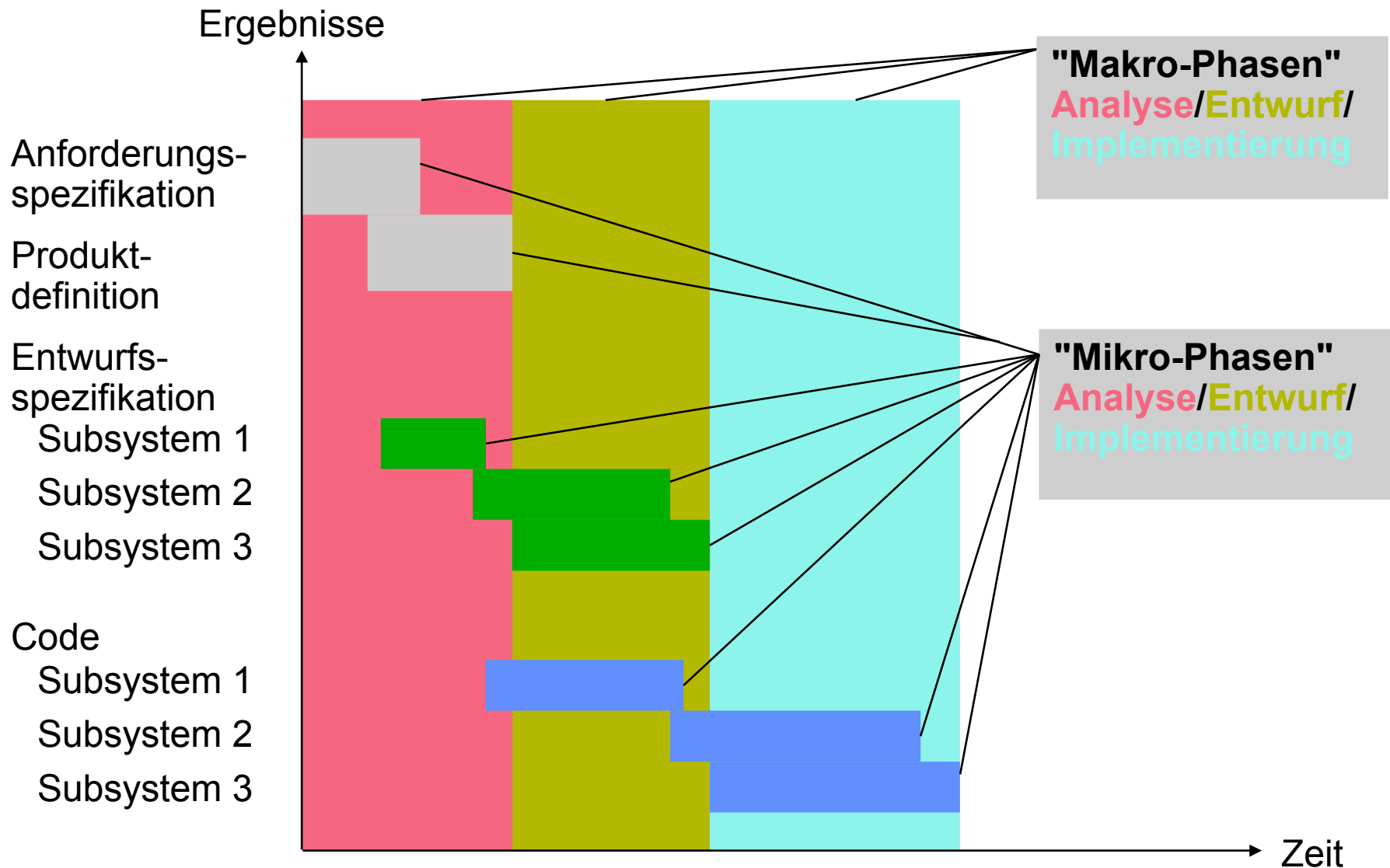
Objektorientiertes Spiralmodell



Spiralmodell vs. evolutionäre Entwicklung

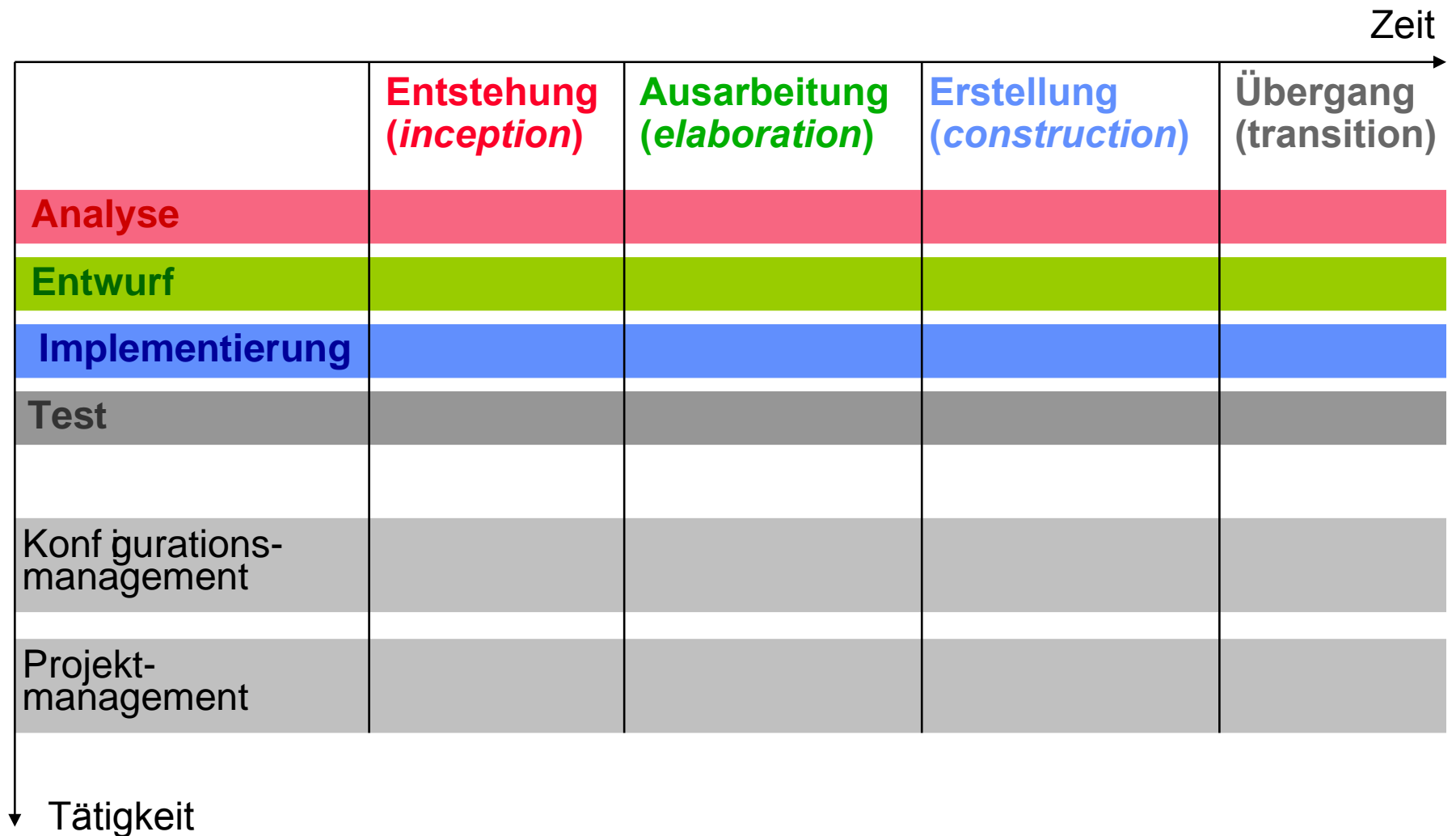
- ▶ Grundidee identisch:
 - Zyklisches Durchlaufen von Entwicklungsaktivitäten
 - Aufeinanderfolgende Prototypen
- ▶ Evolutionäre und agile Entwicklung:
 - Reaktion auf Änderungen ist wichtiger als Verfolgung eines Plans
 - Planung nur für sehr kurze Zeiträume (Tage, Wochen) im voraus
 - Viele, häufige Durchläufe (z.B. Tagesrhythmus)
- ▶ Spiralmodell:
 - Einsetzbar in verschiedener "Strenge"
 - Vorausplanung von Durchläufen
 - Anzahl Durchläufe manchmal schon bei Projektbeginn festgelegt
 - Wenige Durchläufe (z.B. Quartalsrhythmus)
 - Kompromiß zwischen Planbarkeit und Agilität

Parallelität im Entwicklungsprozeß



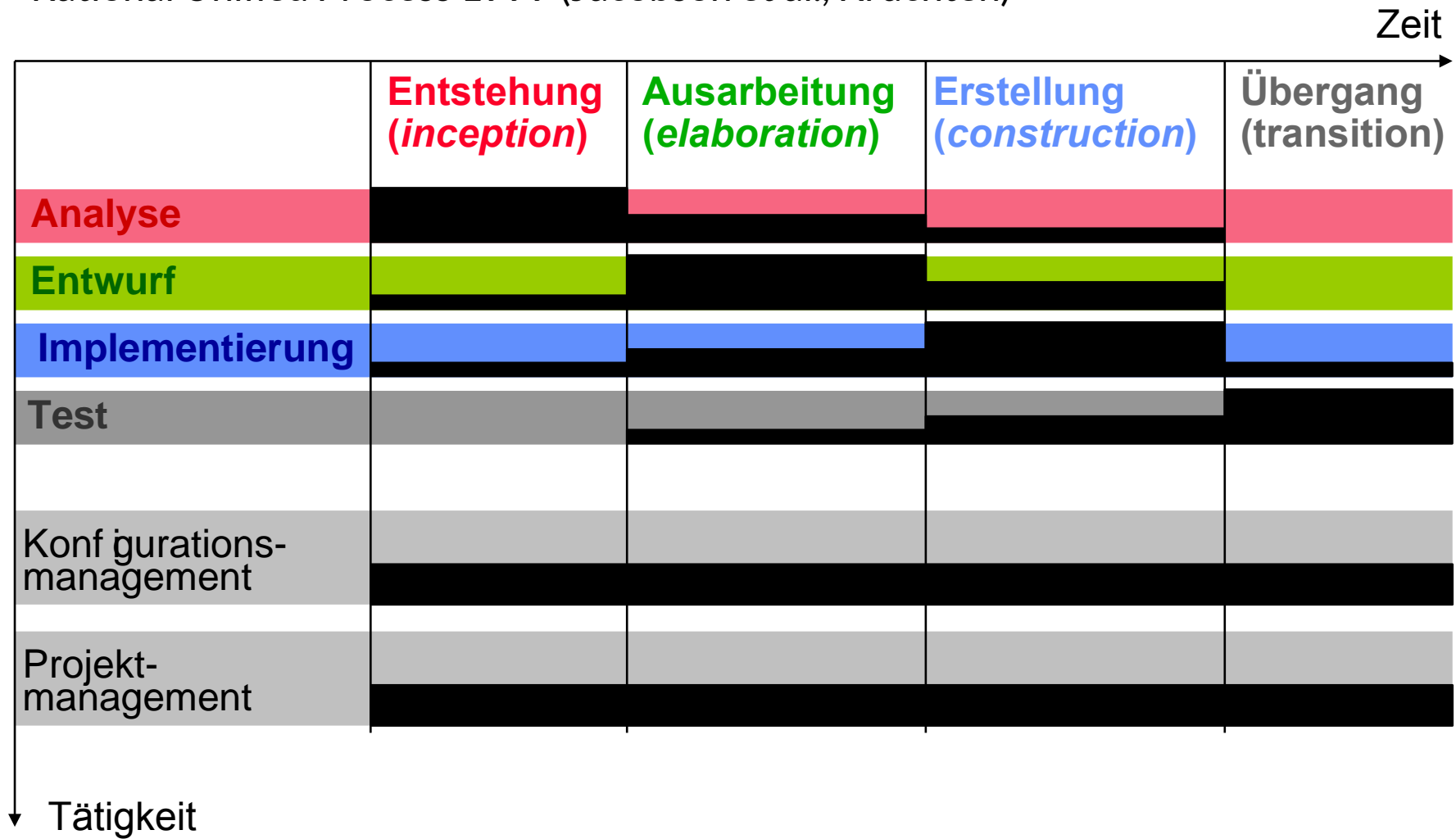
Zweidimensionales Modell

- ▶ Rational Unified Process 1999 (Jacobson et al., Kruchten) mit Mikro- und Makrophasen



Aufwandsverteilung und Schwerpunkte

Rational Unified Process 1999 (Jacobson et al., Kruchten)



Rational Unified Process (RUP)

- ▶ von IBM Rational:

Phases

Core Process Workflows

Business Modeling

Requirements

Analysis & Design

Implementation

Test

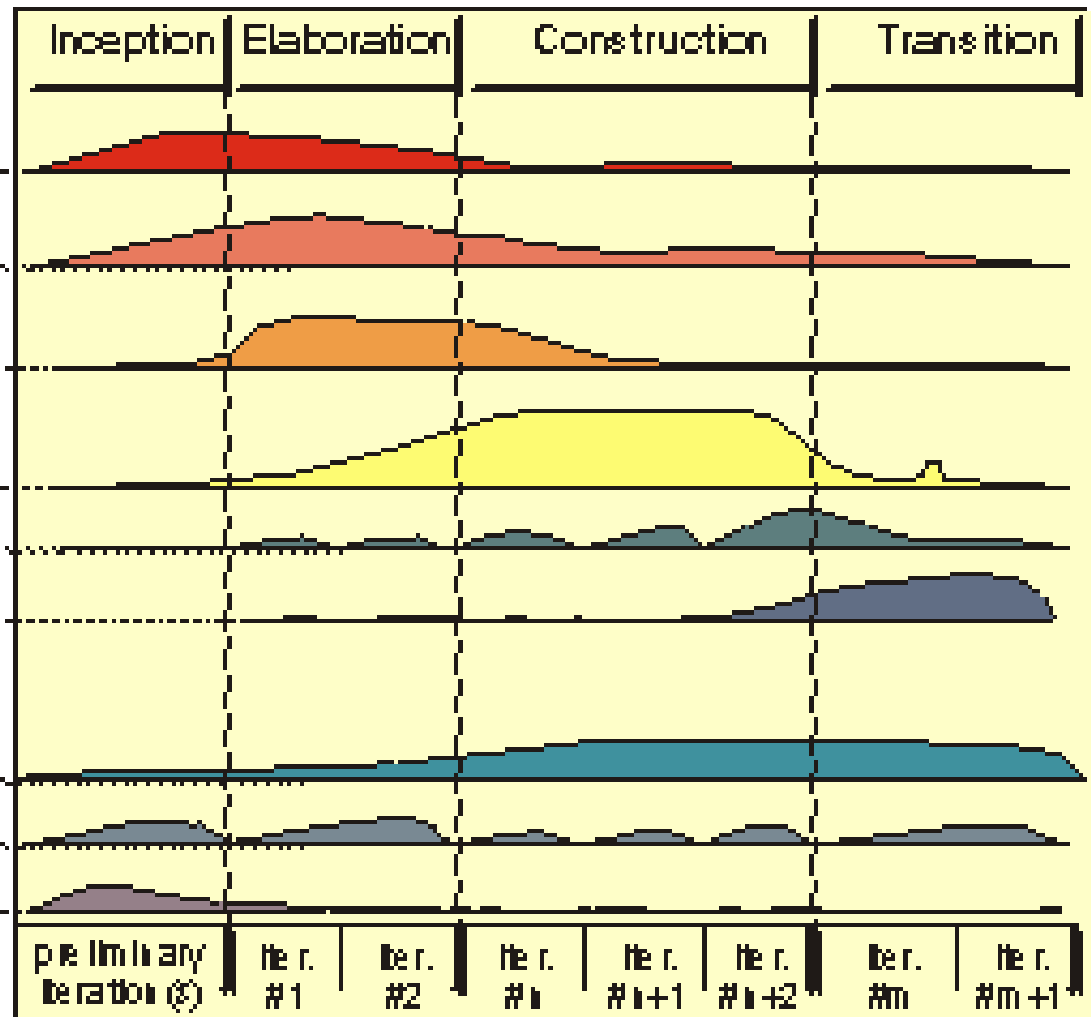
Deployment

Core Supporting Workflows

Configuration & Change Mgmt.

Project Management

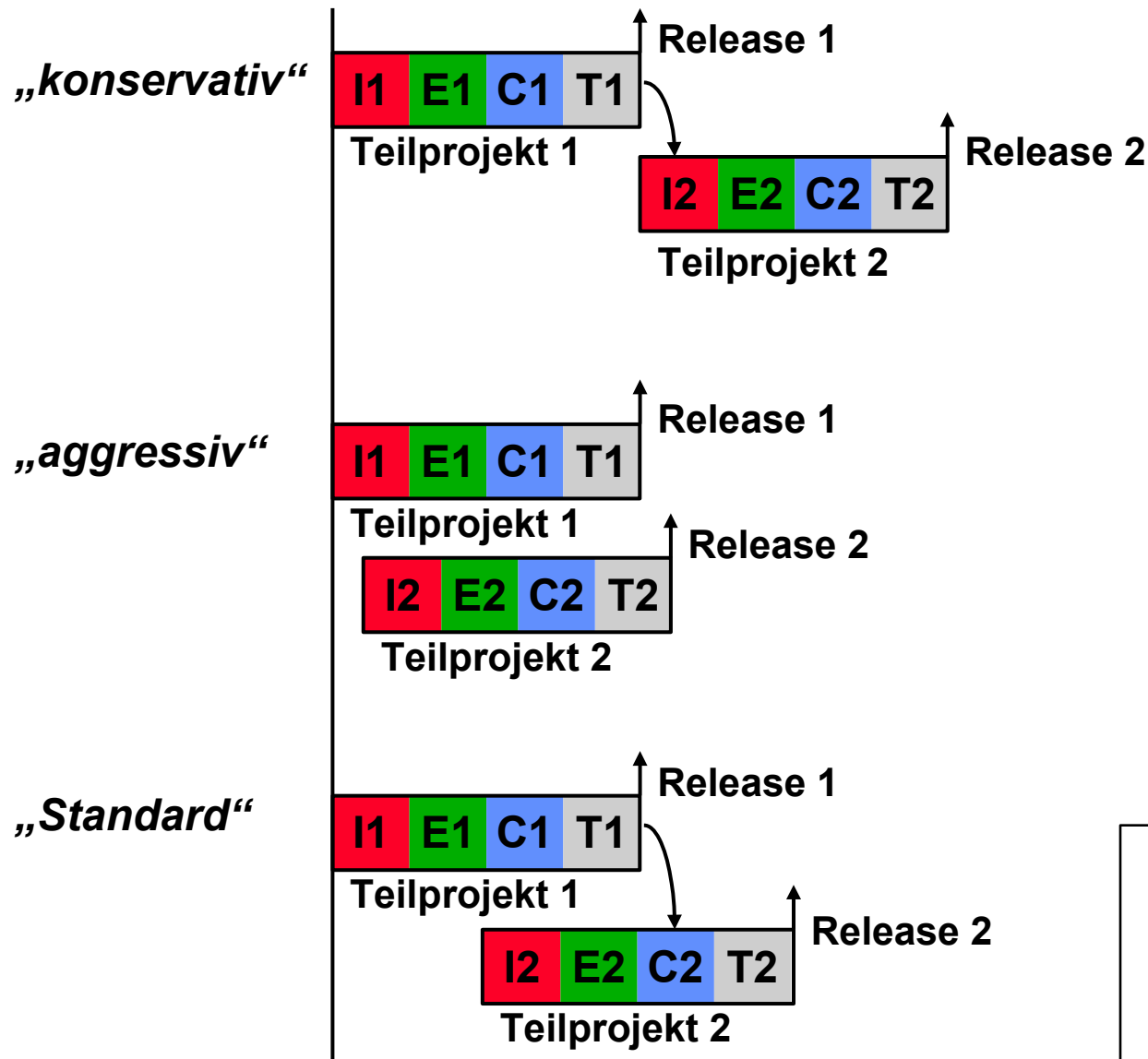
Environment



Iterations

preliminary iteration (0) | iter. #1 | iter. #2 | iter. #i | iter. #i+1 | iter. #i+2 | iter. #m | iter. #m+1

Teilprojekte und Überlappungsgrade



I	Inception
E	Elaboration
C	Construction
T	Transition

Vorgehen im Softwarepraktikum 3. Semester

- ▶ Echte Kunden
- ▶ Vorgehensmodell: V-Modell mit Akzeptanztests
- ▶ Einfache Inkrementalität: Kunde hat einen *Verbesserungswunsch* frei, der erst zu einem späten Zeitpunkt bekanntgegeben wird
- ▶ Intern kann ein inkrementelle Vorgehensmodell gewählt werden

Was haben wir gelernt?

- ▶ Vorgehen nach einem strukturierten Phasenmodell ist gewöhnlich besser als ad-hoc Vorgehen
- ▶ Realistische Vorgehensmodelle sind iterativ und inkrementell
- ▶ Der Ingenieur misst, entwirft, validiert und verbessert

Thus it will be seen that *engineering is a distinctive and important profession*. To some even it is the topmost of all professions. However true that may or may not be to-day, certain it is that some day it will be true, for the reason that engineers serve humanity at every practical turn.

Engineers make life easier to live--easier in the living; their work is strictly constructive, sharply exact; the results positive.

Charles M. Horton. Opportunities in Engineering. 1920, by Harper & Brothers

<http://www.gutenberg.org/ebooks/24681>

.. Not a profession outside of the engineering profession but that has its moments of wobbling and indecision--of faltering on the part of practitioners between the true and the untrue. Engineering knows no such weakness. *Two and two make four. Engineers know that.* Knowing it, and knowing also the unnumbered possible manifoldings of this fundamental truism, engineers can, and do, approach a problem with a certainty of conviction and a confidence in the powers of their working-tools nowhere permitted men outside the profession.

Charles M. Horton. Opportunities in Engineering. 1920, by Harper & Brothers

<http://www.gutenberg.org/ebooks/24681>

- ▶ Die deutschen Folien der Softwaretechnologie-Vorlesung stammen zu Teilen aus den Jahren 2000-2003 (Prof. Dr. Heinrich Hussmann, jetzt LMU München). Used by permission.