

# Teil V: Projektmanagement

## 50 Projektplanung

1

Prof. Dr. rer. nat. Uwe Aßmann  
Institut für Software- und  
Multimediatechnik  
Lehrstuhl Softwaretechnologie  
Fakultät für Informatik  
TU Dresden  
Version 14-1.0, 20.07.15

- 1) Projektmanagement
- 2) Vorgehensmodelle



# Einschreiben ins Praktikum bis 3.8.2015

2

- ▶ über [www.jexam.de](http://www.jexam.de)
- ▶ Einschreibung befristet, da Gruppen geplant werden müssen!
- ▶ **Wer sich nicht einschreibt, kann nicht teilnehmen**

# 50.1 Projektmanagement

und eines  
Projektes

3

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

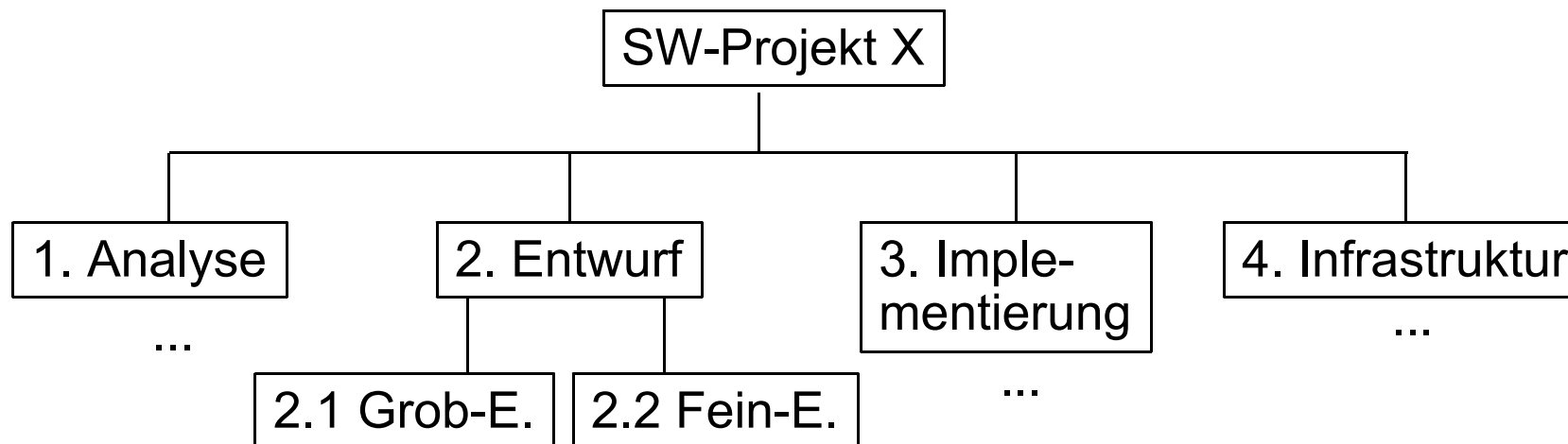
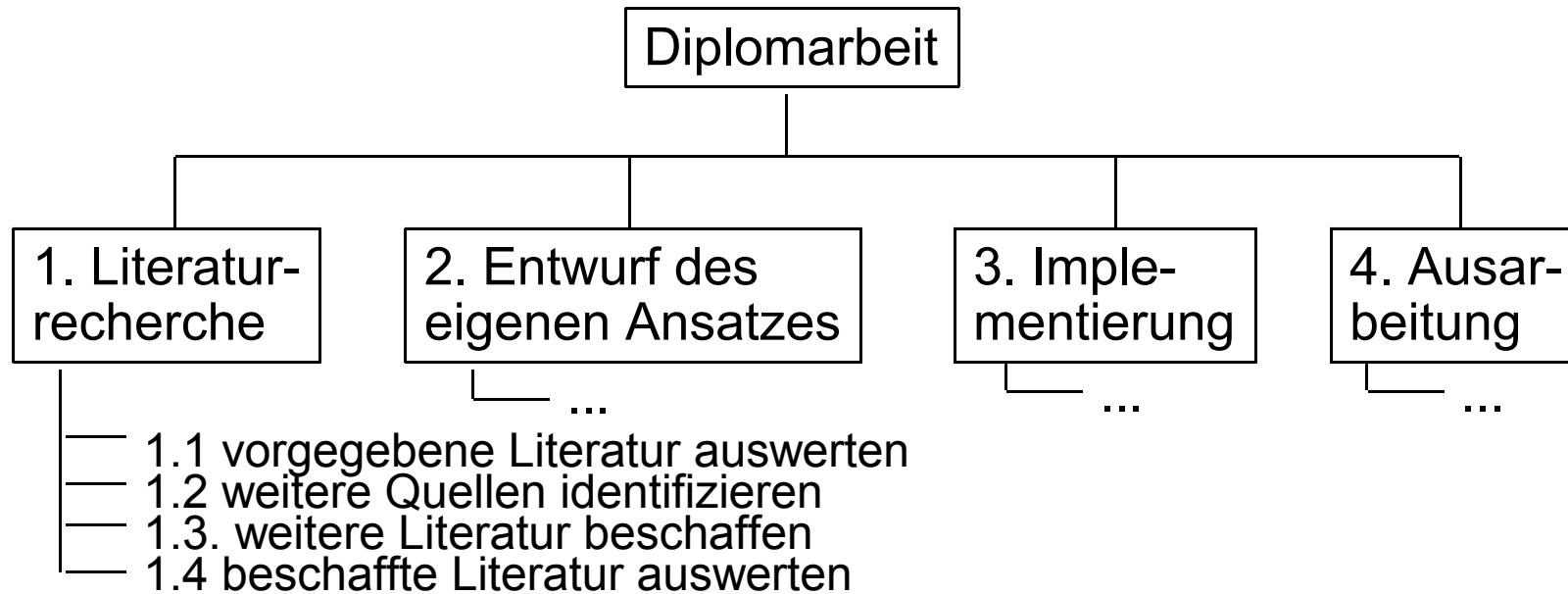
Carl Hilty, 28.02.1833 - 12.10.1909

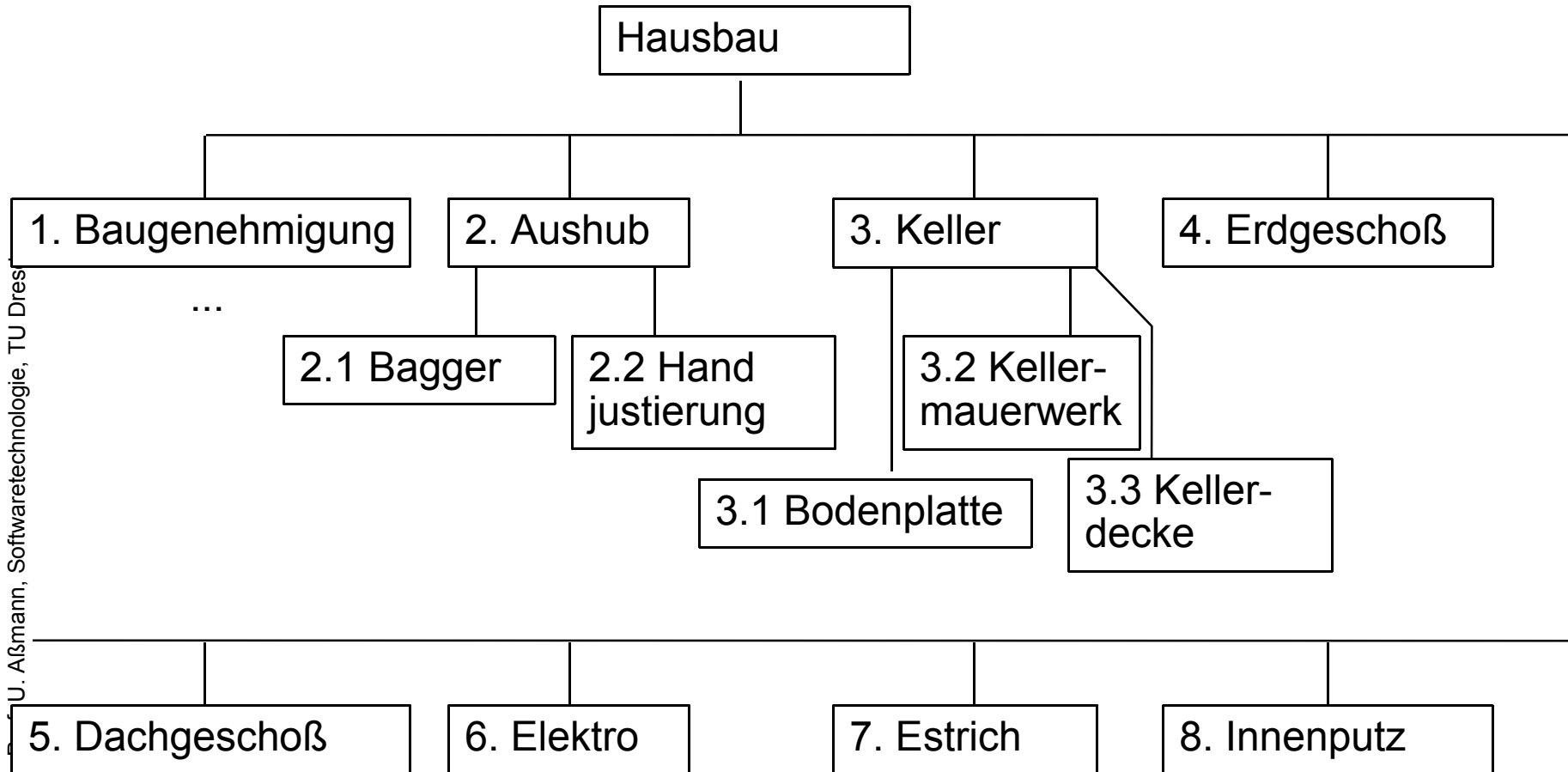
Schweizer Richter und Staatsrechtler, Buchautor und christl. Staatsrechts-Philosoph

Seine Bücher beeinflussten auch K. Adenauer

# Projektstruktur (“Work Breakdown Structure”): Beispiele

4





# Aufwandsschätzung

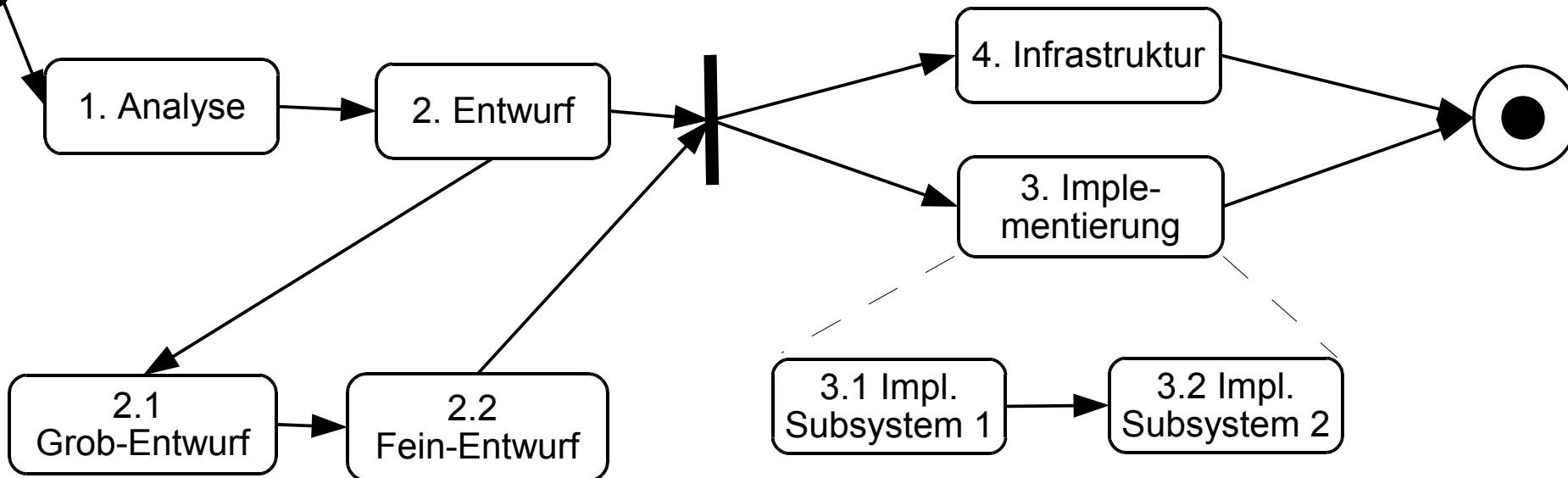
6

- ▶ 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

7

- ▶ 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

8

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

9

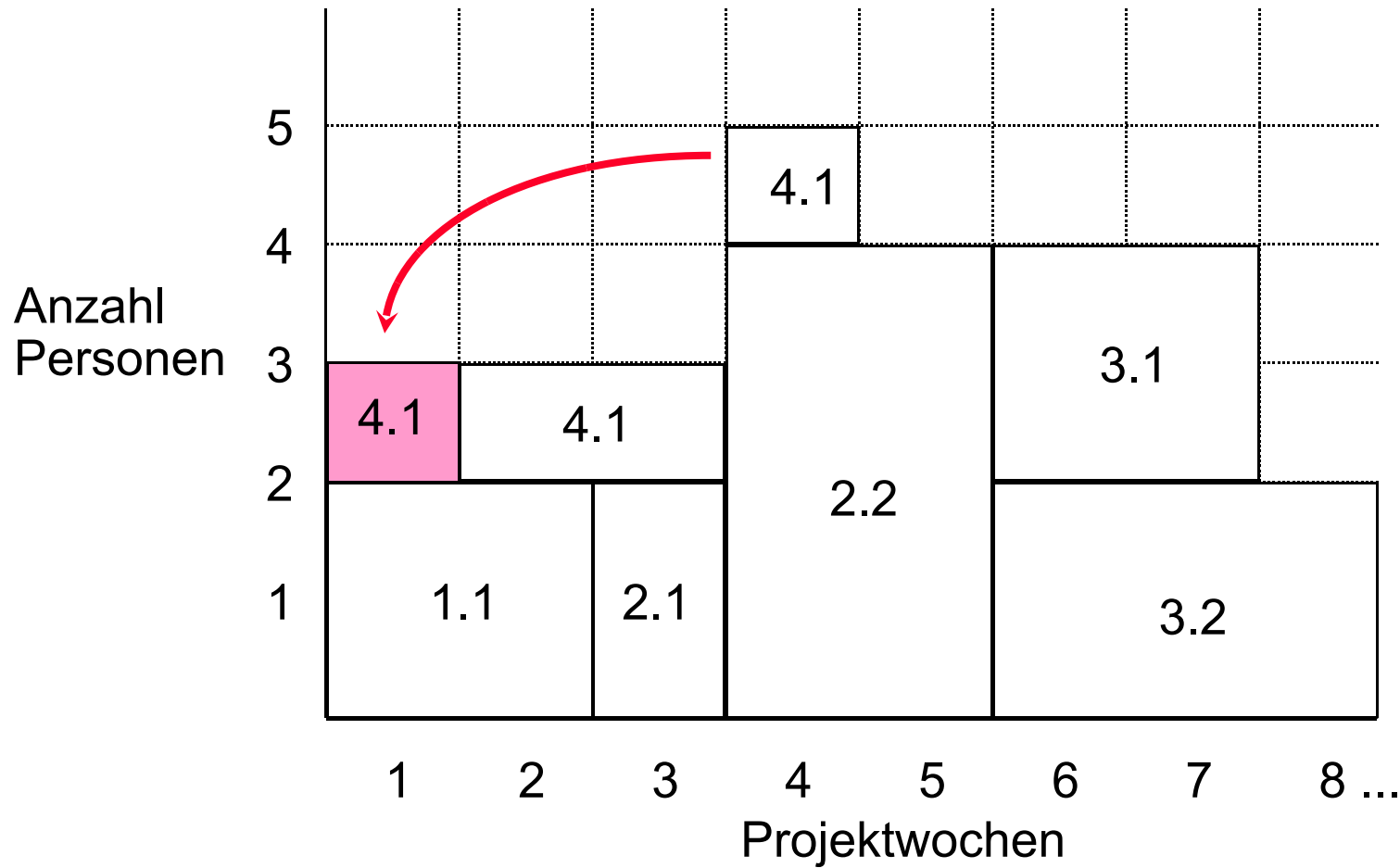
Projektwochen

Arbeitspaket	1	2	3	4	5	6	7	8 ...
1.1 Baugenehmig.	■							
2.1 Aushub			■					
2.2 Keller				■				
3.1 Erdgeschoß					■			
3.2 Dachgeschoß...						■		
14.1 Hausanschluß				■				

# Ressourcenplanung

10

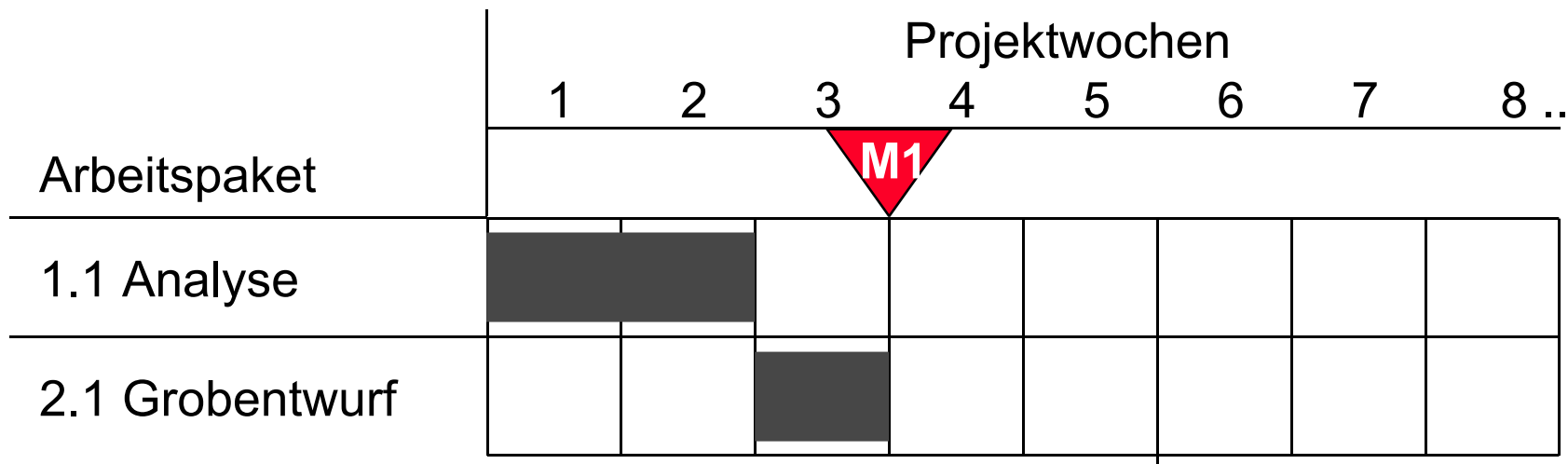
- ▶ Umplanung mit dem Ziel: Anpassung an vorhandene Ressourcen
- ▶ *Packen* in Flächen über Anz. Personen und Projektwochen



# Meilensteine

11

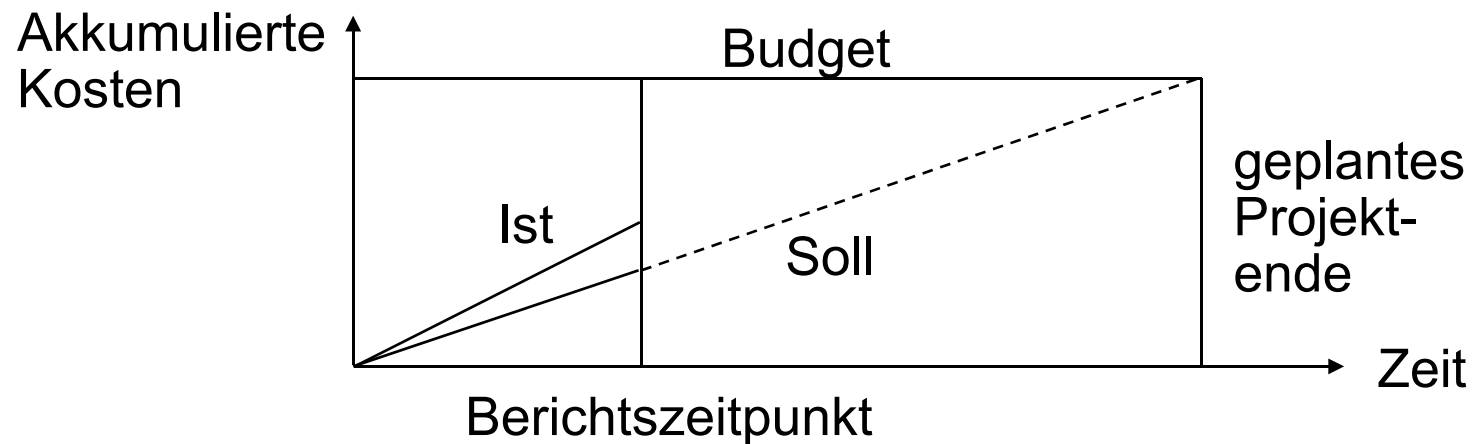
- ▶ Ein *Meilenstein* ist ein klar definiertes Zwischenresultat, an Hand dessen der Projektfortschritt beurteilt werden kann.
- ▶ Beispiele:
  - "Anforderungsspezifikation zusammen mit Auftraggeber verabschiedet"
  - "Erster Prototyp lauffähig"
  - Schlechtes Beispiel: "Code zu 50% fertig"
- ▶ Meilensteine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



# Projektverfolgung

12

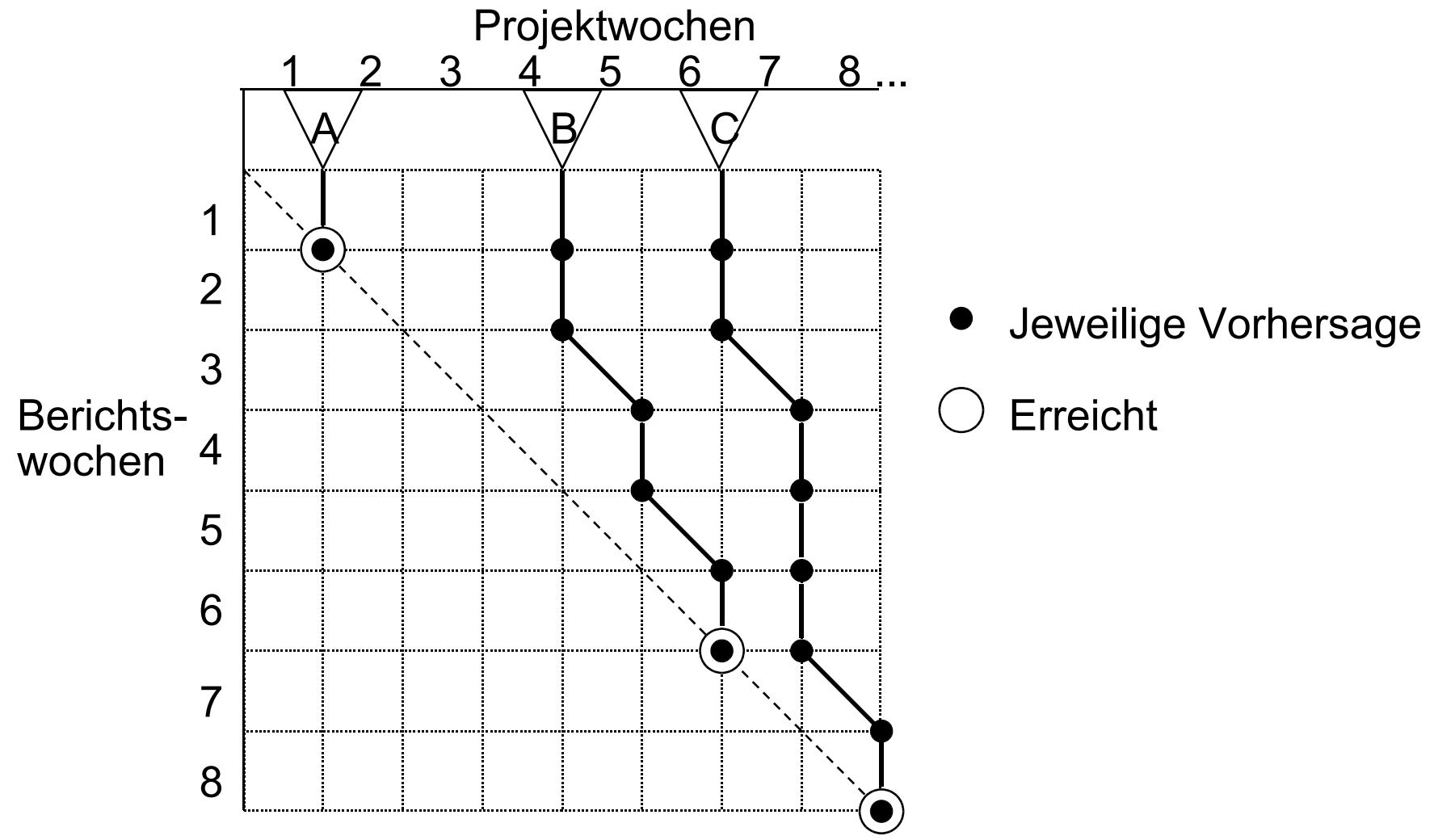
- ▶ Das Projektmanagement muß ein "Frühwarnsystem" für eventuelle Probleme betreiben (Projektverfolgung).
- ▶ Informationsquellen:
  - Laufende (z.B. wöchentliche) Management-Berichte
  - Arbeitszeit-Kontierung
  - Resultate (*deliverables*)
- ▶ Rückkopplung zum Projektteam
  - Regelmäßige Projektbesprechungen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



# Meilenstein-Trendanalyse

13

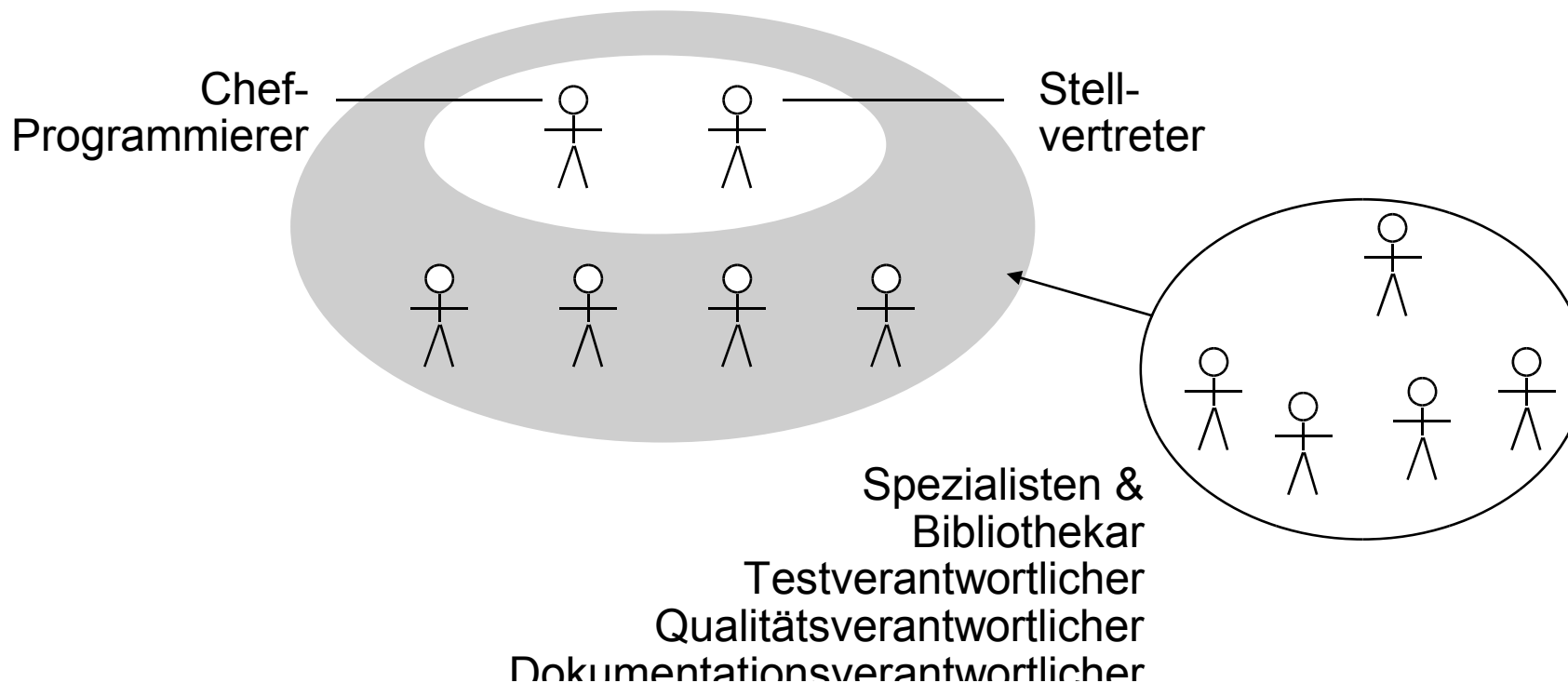
- ▶ Anhand jedes Managementberichts sagt das Management die Meilensteine neu voraus



# Teamzusammenstellung (Staffing)

14

- ▶ 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

15

- ▶ 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

# Typische Gliederung eines Ergebnisprotokolls

16

- ▶ 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*)

17

- ▶ 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. GitHub, Mantis.org)

# Aufgabenmanagement-Werkzeuge

18

- ▶ **Einzelaufgaben (Aktivitäten)** werden in ein AufgabenmanagementäSystem eingetragen (“ticket system”, “issue management system”)

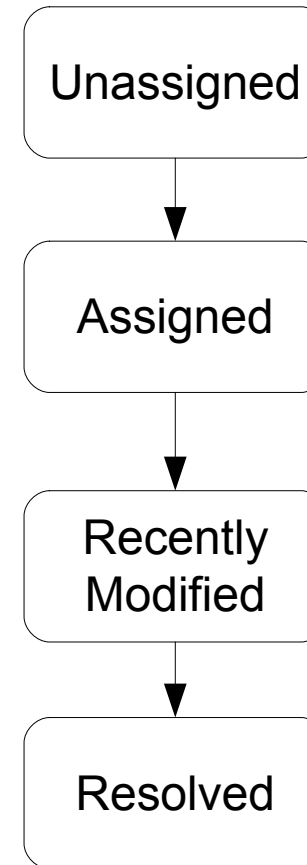
Werkzeug	Status	Webadresse
Bugzilla	Mozilla (OSS)	www.bugzilla.org
<b>Mantis</b>	OSS	http://www.mantisbt.org/
<b>JIRA</b>	Atlassian	http://atlassian.com/software/jira
codeBeamer	IntLand Software	http://intland.com/products/codebeamer/overview/
<b>RedMine</b>	OSS	http://en.wikipedia.org/wiki/Redmine
Team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

[http://en.wikipedia.org/wiki/Comparison\\_of\\_issue\\_tracking\\_systems](http://en.wikipedia.org/wiki/Comparison_of_issue_tracking_systems)

# Aufgabenmanagement (Fehler und Änderungen) mit Mantis

19

- ▶ 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]

20

The screenshot shows the MantisBT web interface. At the top, there is a navigation bar with links like 'Main', 'My View', 'View Issues', 'Change Log', 'Roadmap', 'Docs', 'Wiki', and 'Billing'. Below this is a search bar and filter options. The main content area displays a table of issues with columns for ID, Category, Severity, Status, Updated, and Summary. The table is sorted by ID in descending order, showing issues from 0005069 to 0005072. The interface also includes a sidebar with project navigation and a top navigation menu.

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	asdasdsad
v	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 associação 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 (mow)	2009-01-26	Alle Schiffe



# Überblick über Zustände der Aufgaben

21

Prof. U. Aßmann, Softwaretechnologie, TU Dresden

The screenshot displays the MantisBT web interface for a project named 'Demo'. The page is titled 'My View - MantisBT DEMO site' and shows the current date and time as '2009-01-28 14:21 EST'. The user is logged in as 'Anonymous'. The interface includes a search bar, a navigation menu, and three main sections for task lists.

**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	sdcd 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



# KM-Werkzeuge

22

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

# Konkrete Aufgaben zur Vorbereitung des Praktikums

23

- ▶ Einlesen:
- ▶ GitHub
- ▶ git



# 50.2 Vorgehensmodelle (Phasenmodelle)

---

---

24





- ▶ Softwaretechnologie für Einsteiger. PEARSON, 2014, 2. Auflage

## ***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?

26

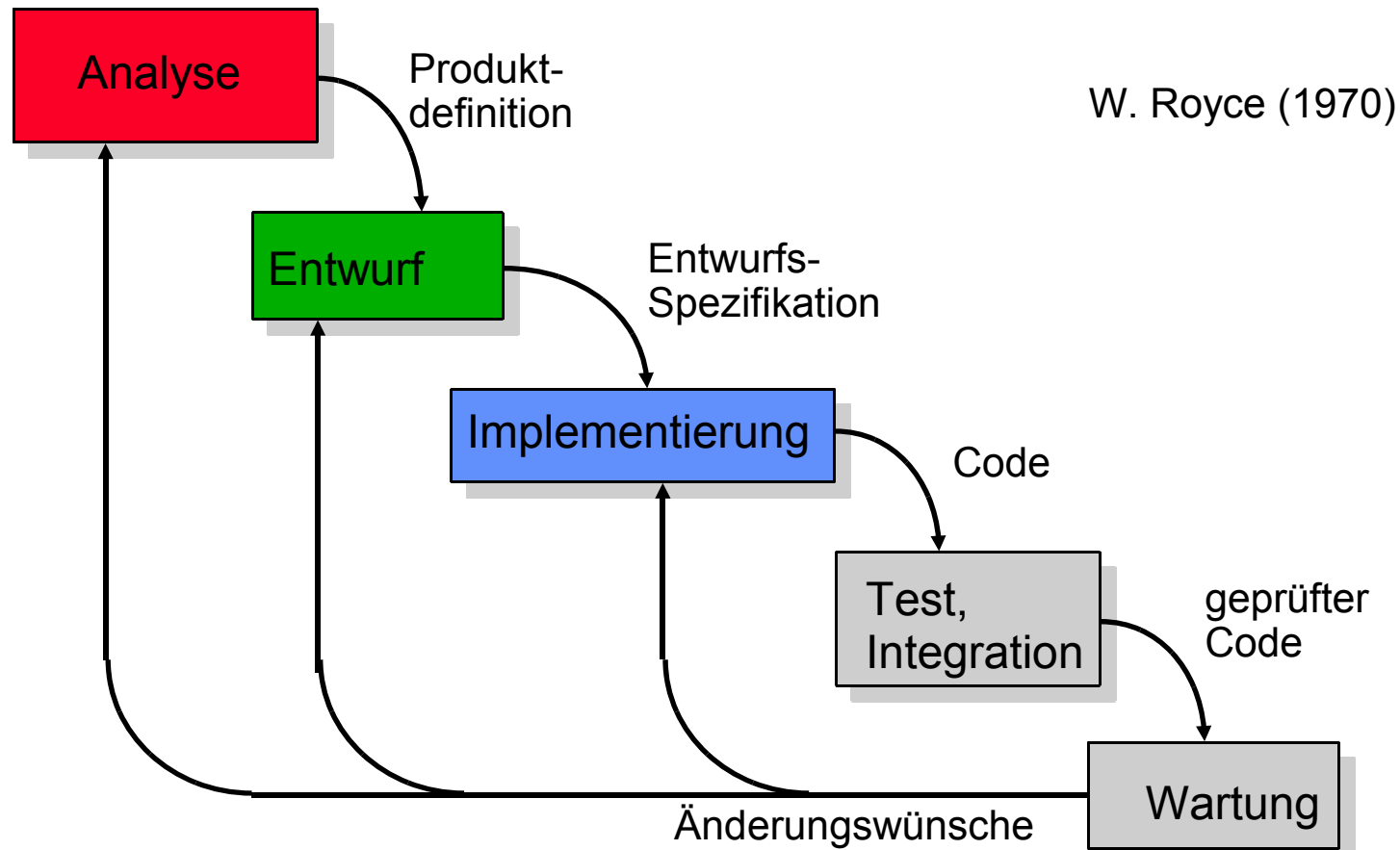
- ▶ Ad hoc
- ▶ Es lief schon oft schief...
  - Denver International Airport, Krise 1993-95
  - Hamburger Güterbahnhof 1995
  - Toll Collect 2004
  - Fiscus 2004
- ▶ Gibt es nicht irgendwelche Hilfen, strukturiert vorzugehen?

# Vorgehen nach einem “Phasenmodell”

27

- ▶ **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)

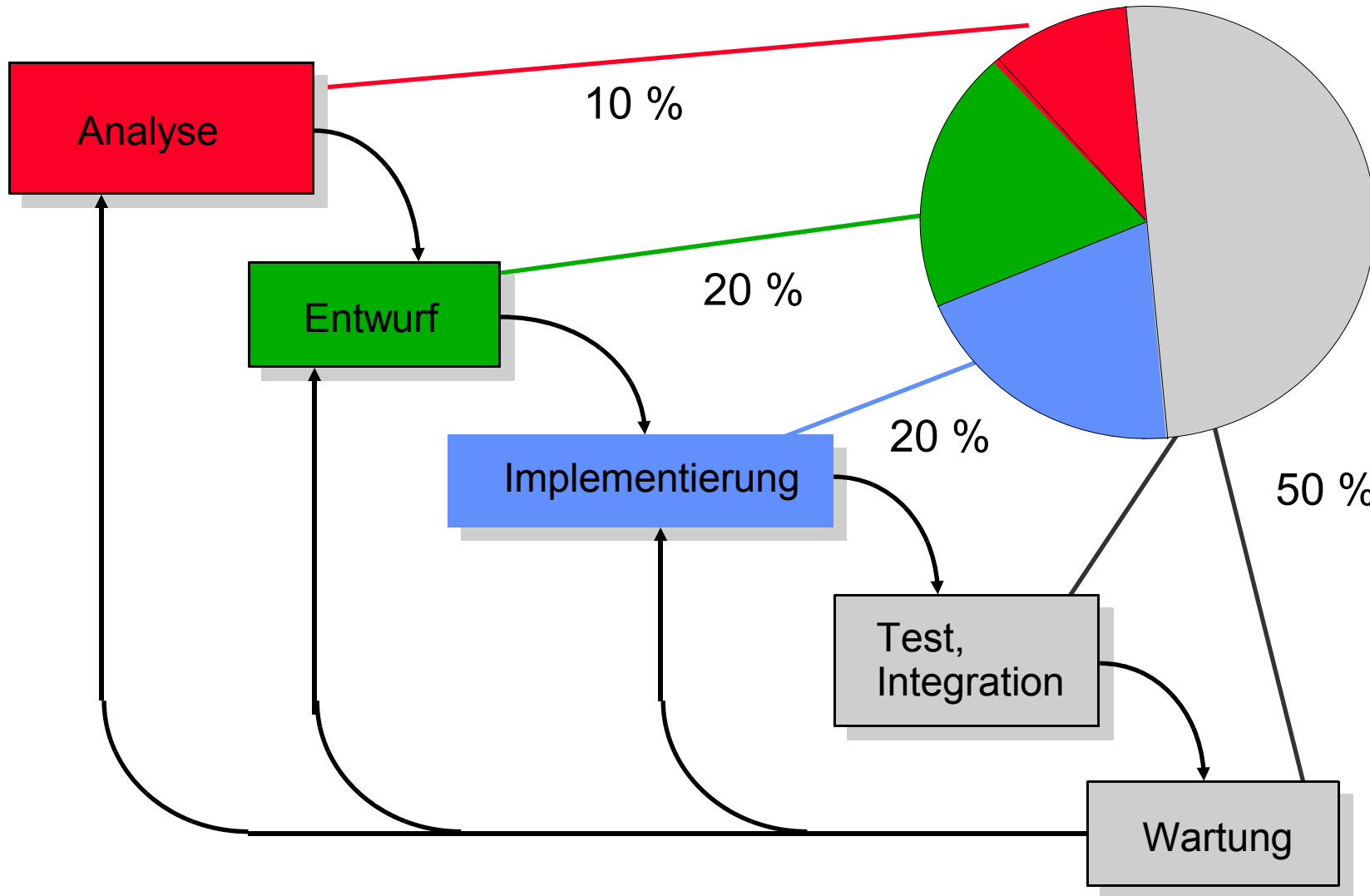
# Wasserfall-Modell (mit Iterationen)



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

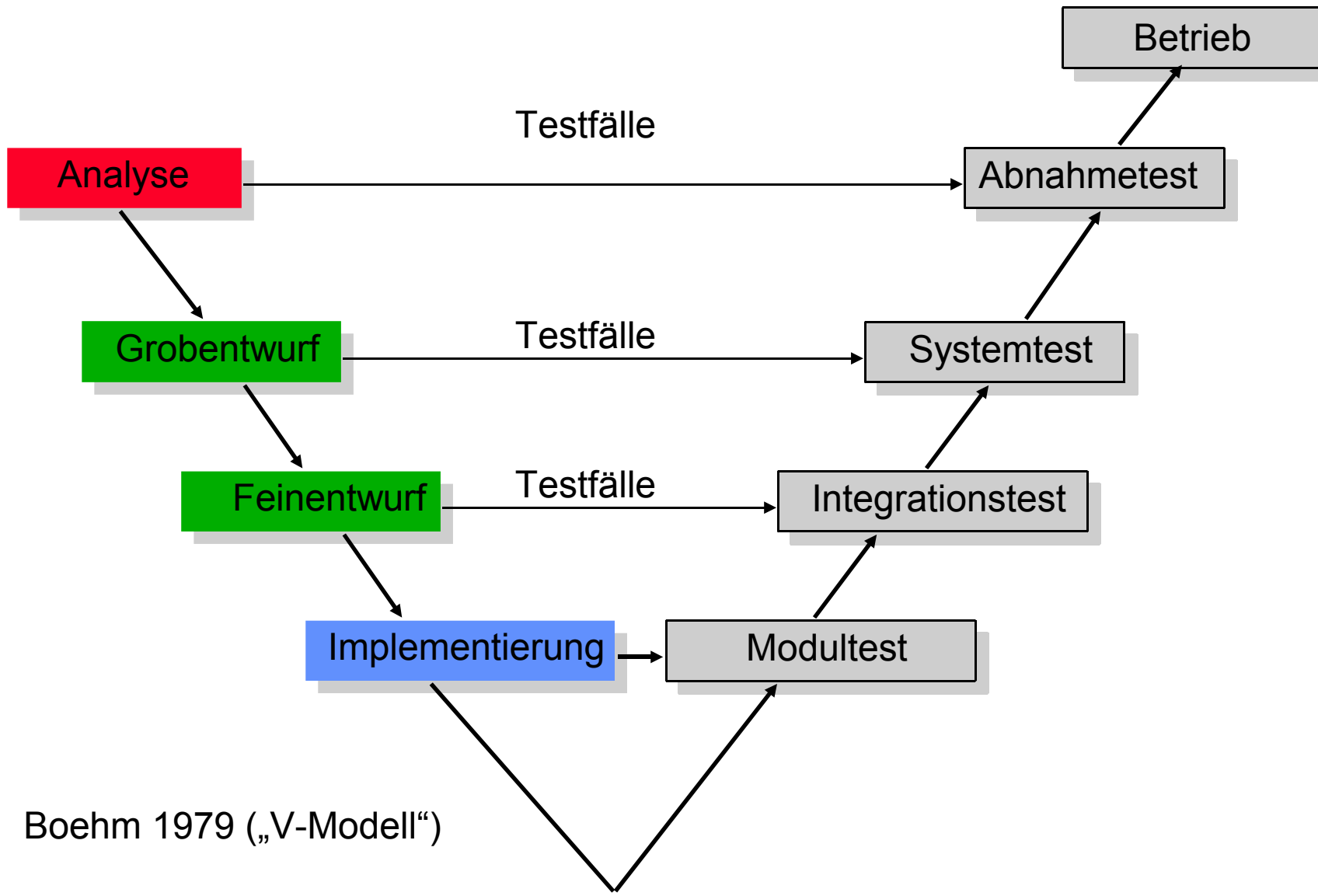
# Ungefähre Verteilung des Arbeitsaufwandes

29



# Qualitätssicherung im einfachen V-Modell

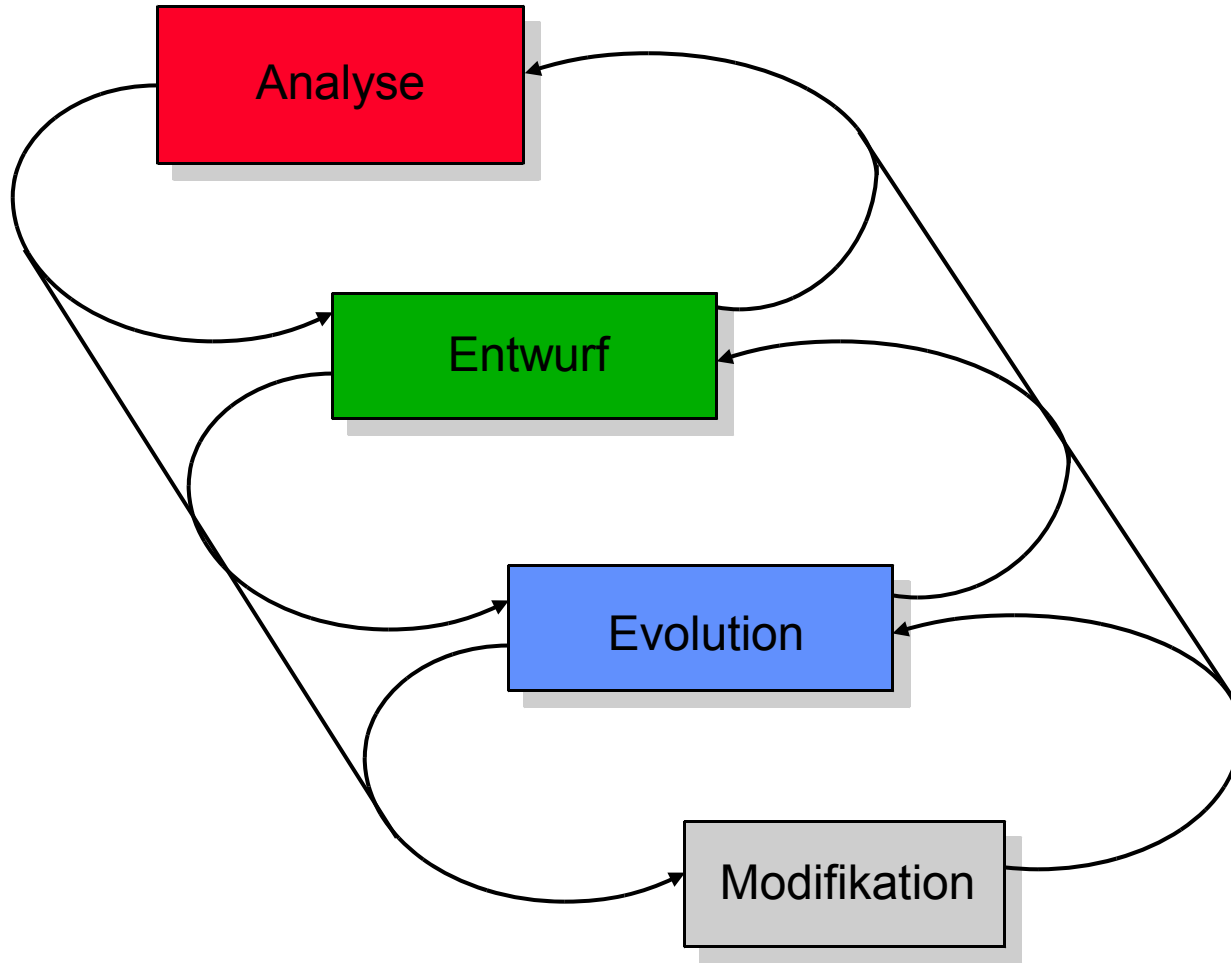
30



Boehm 1979 („V-Modell“)



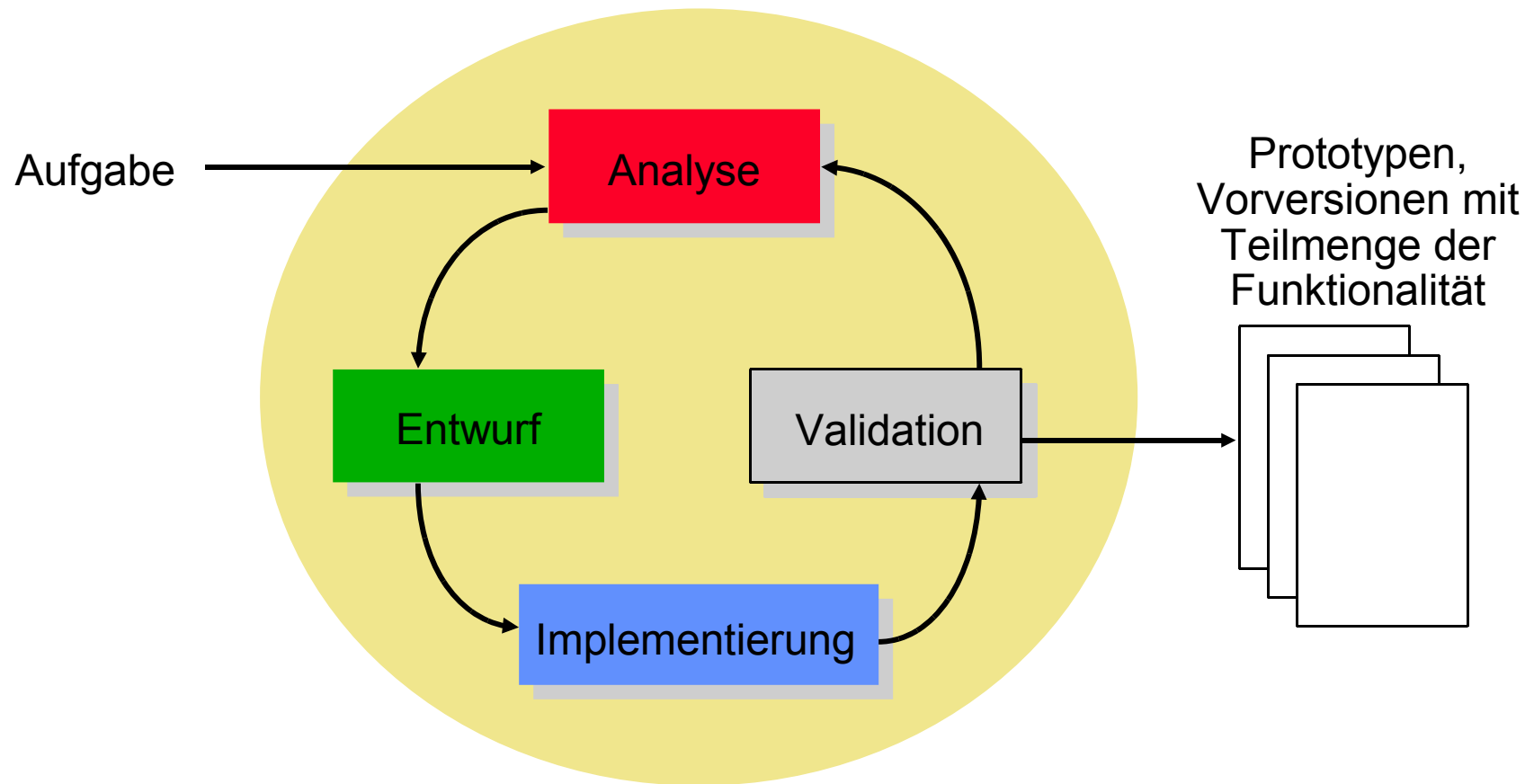
# Inkrementelle (evolutionäre, agile) Entwicklung



# Evolutionäre Entwicklung

32

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





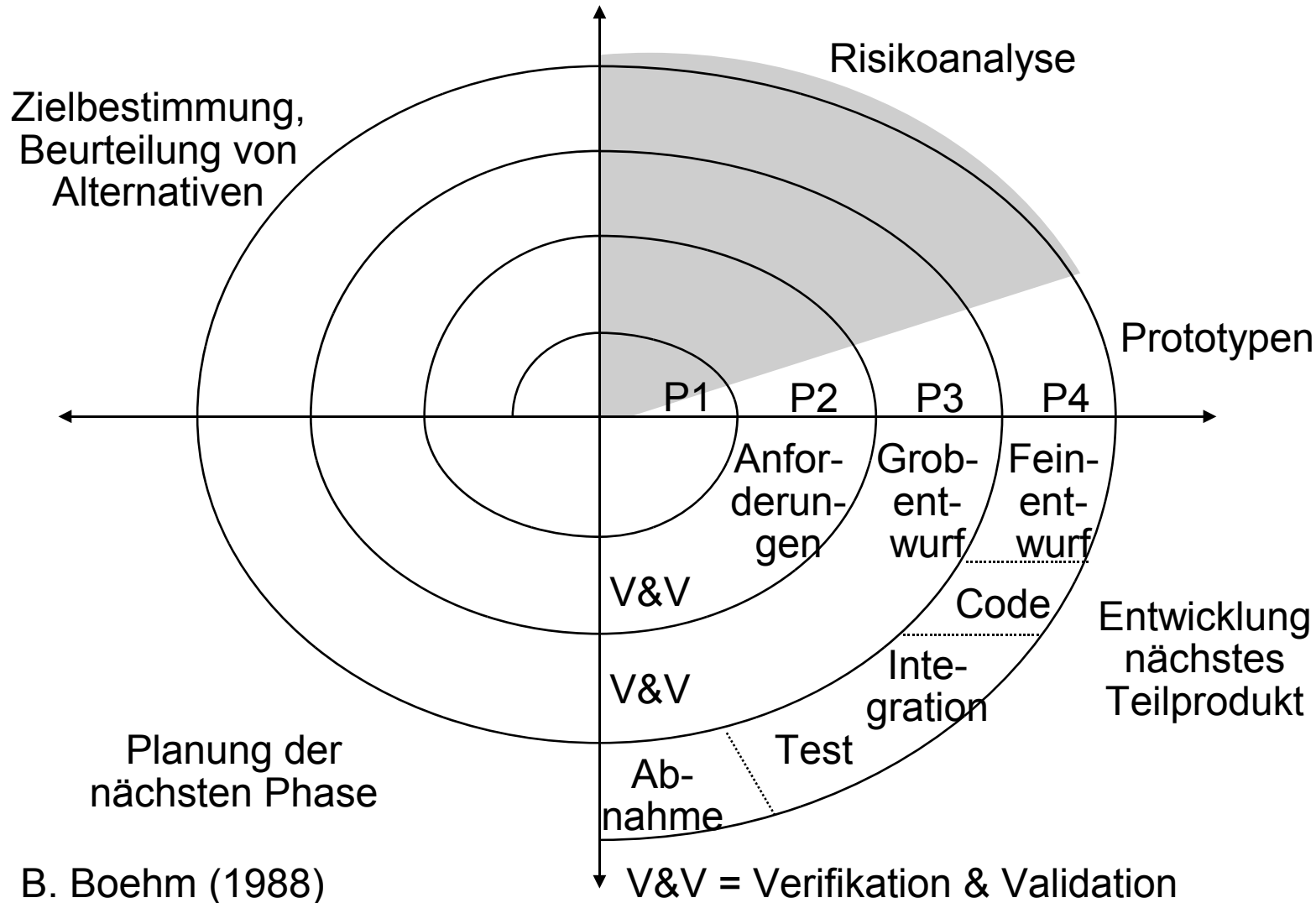
# eXtreme Programming (XP)

33

- ▶ Kontrovers diskutierete 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](http://www.agilemanifesto.org)):
  - weitere Ansätze, z.B. Crystal, **Scrum**

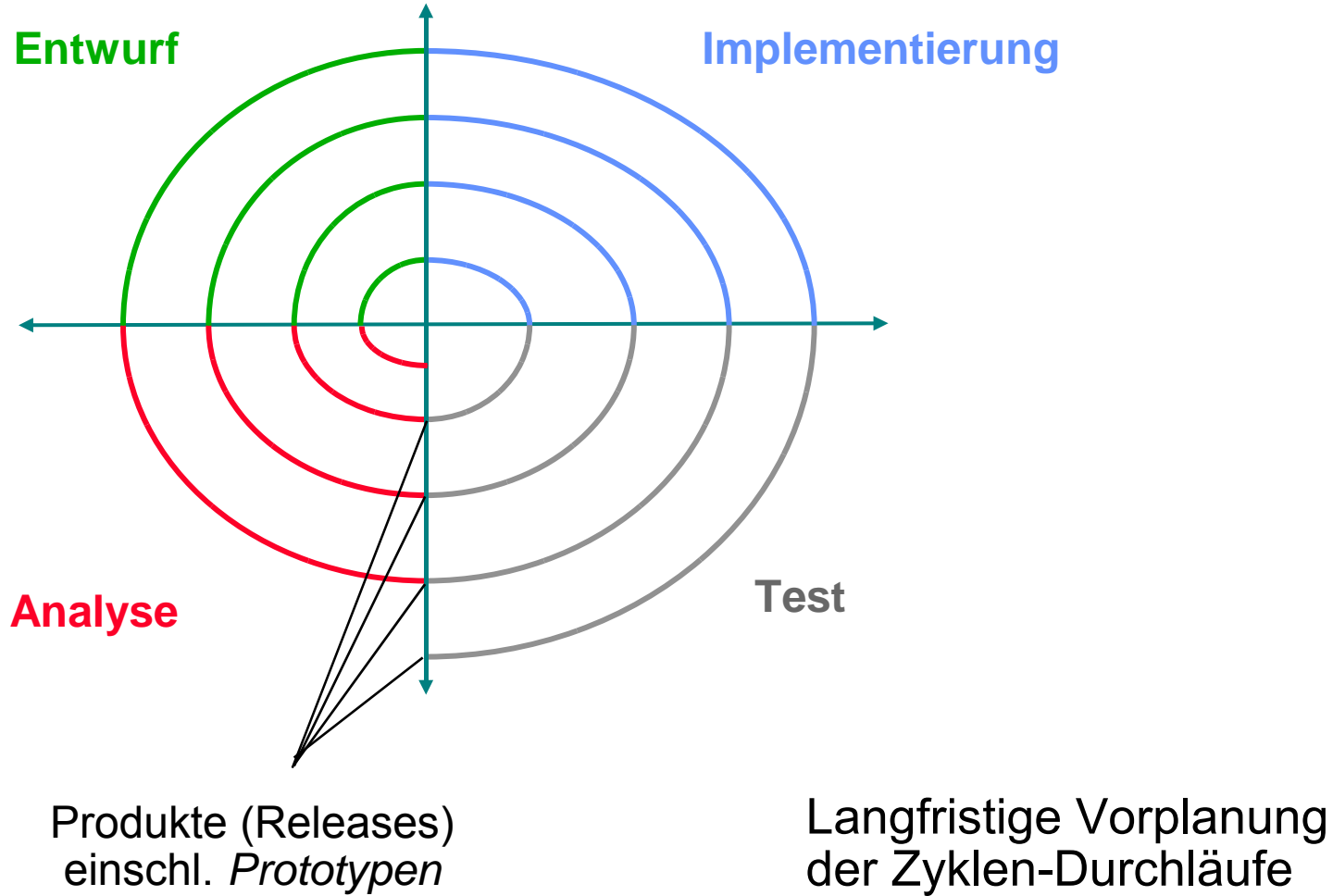
# Spiralmodell

34



# Objektorientiertes Spiralmodell

35



# Spiralmodell vs. evolutionäre Entwicklung

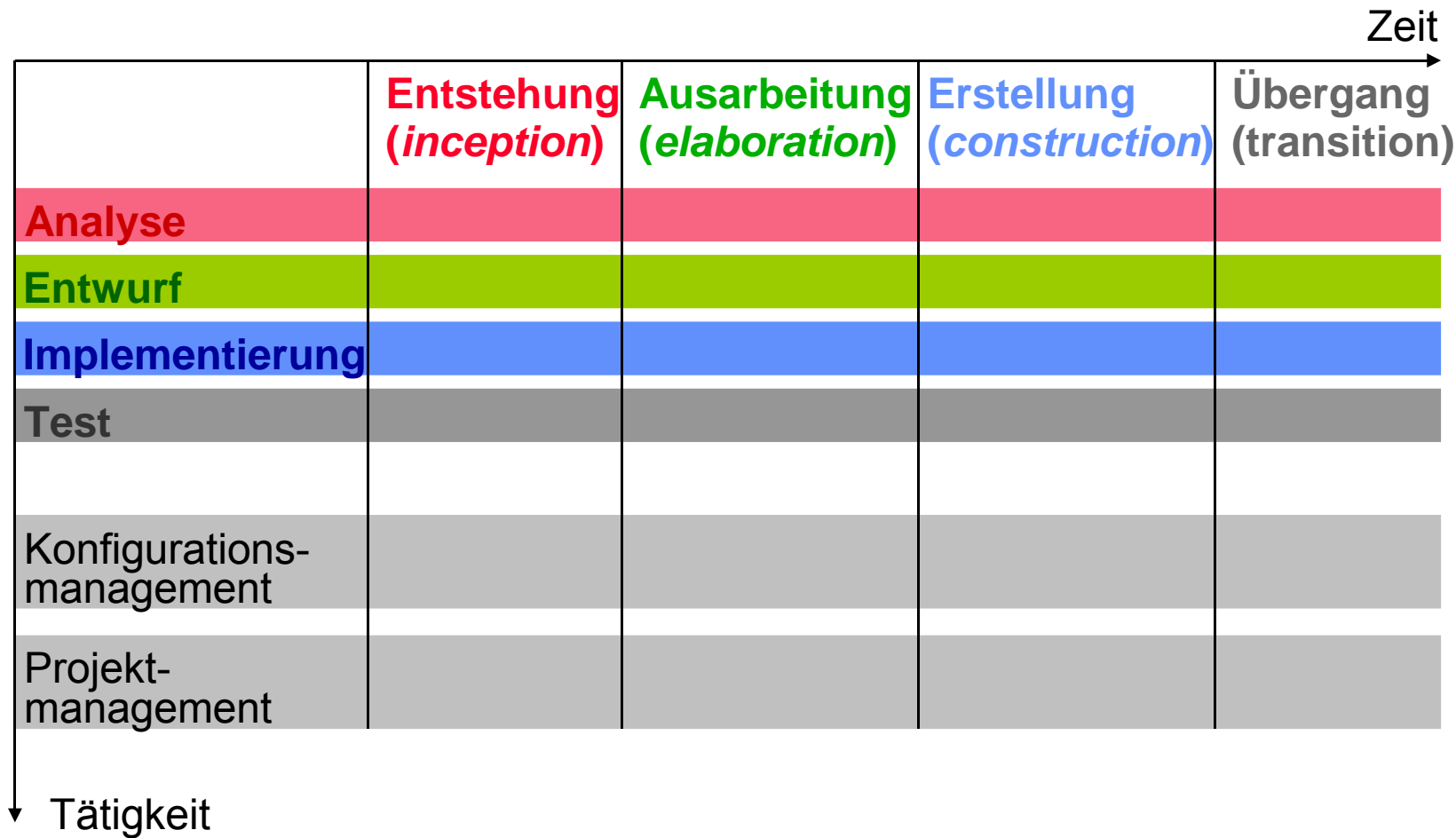
36

- ▶ 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

# Zweidimensionales Modell

37

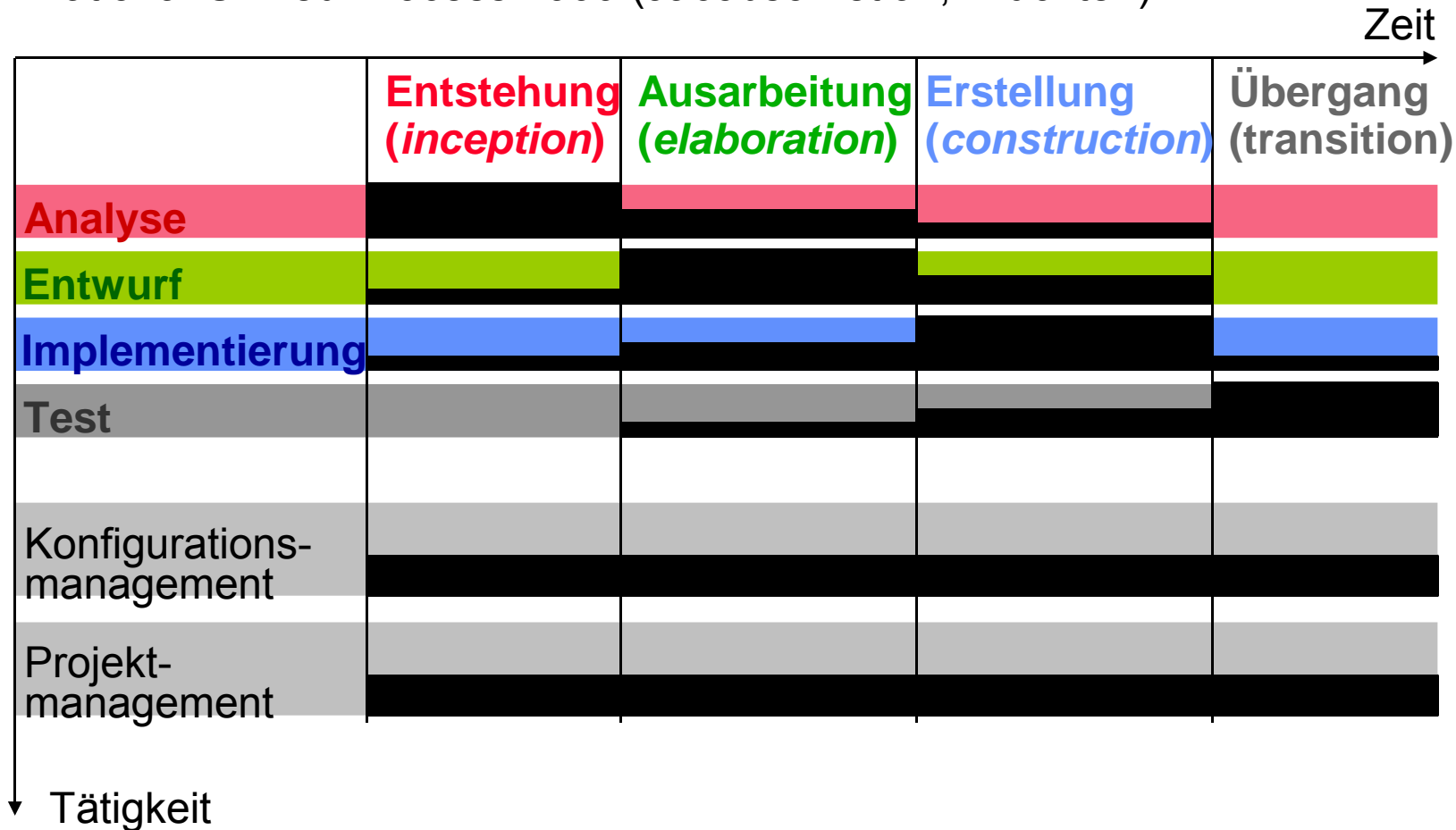
- ▶ Rational Unified Process 1999 (Jacobson et al., Kruchten) mit Mikro- und Makrophasen



# Aufwandsverteilung und Schwerpunkte

38

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



# Vorgehen im Softwarepraktikum 3. Semester

---

39

- ▶ Echte Kunden
- ▶ Vorgehensmodell: modellbasiert, evolutionär, mit Meilensteinen und SCRUM-Ideen
- ▶ Einfache Inkrementalität: Kunde hat einen *Verbesserungswunsch* frei, der erst zu einem späten Zeitpunkt bekanntgegeben wird

# Was haben wir gelernt?

40

- ▶ 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>



# Referenz

41

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