

Objektorientierte Analyse und Design

Mit UML und dem Unified Process

Vorbemerkungen

- UML ist nur eine Notation,
 - es reicht nicht, 100% syntaktisch korrekte UML Diagramme zu erstellen
 - wichtig ist es, in “Objekten” zu denken
 - und UML richtig *anzuwenden*

Verantwortlichkeiten

- Das wichtigste im OOAD
 - Zuordnen von Verantwortlichkeiten

Analyse? Design?

- Analyse:
 - Ermittlung/Untersuchung des Problems/der Anforderungen
 - Vokabular der Problemdomäne
 - “Was soll das System leisten?”, “Do the right thing!”
- Design:
 - Erstellung einer *konzeptuellen Lösungsskizze*, die die Anforderungen erfüllt
 - Vokabular der Lösungsdomäne
 - “Do the thing right!”

Unified Process

Unified Process

Was macht den Unified Process aus?

Unified Process

- Iteratives und inkrementelles Vorgehen
 - Schrittweise Erweiterung und Verfeinerung des Systems
 - In jeder Iteration werden alle Aktivitäten (Analyse, Design, Implementierung, Test) durchgeführt
 - Am Ende jeder Iteration liegt ein funktionsfähiges (aber unvollständiges) System vor
 - Neue Features oder Verbesserung vorhandener

Unified Process

- Kein *rush to code*: Analyse und Design finden statt
- kein Experimentieren oder *throw-away prototyping*
- kein Warten, bis alle Anforderungen vollständig und definitiv geklärt sind
- kein Warten, bis das Design perfekt ist
- Embrace Change: Änderungswünsche werden nicht bekämpft, sondern “Treiben” den Fortschritt
 - “Ja, das wollte ich, aber...”
- kein “feature creep” Prozeß (abwägen!)

Unified Process

- *early feedback*:
 - Prüfung der Anforderungen: Feedback von Anwendern
 - Prüfung des Designs: Lauffähiges System
- Kritische Designentscheidungen/Architektur können *früh* statt *spät* korrigiert werden
- Jede Iteration ist ein Zyklus aus *build-feedback-adapt*
- Iterationen sind *timeboxed*: 2 bis 6 Wochen
- Über die Zeit nähert sich das System dem “richtigen” System an

Unified Process

- Anwendungsfall gesteuert
 - Verfolgbarkeit
- Iterativ und Inkrementell
 - kleine Schritte
 - geringes Risiko
 - kleine Rückschläge

Vergleich Vorgehensmodelle

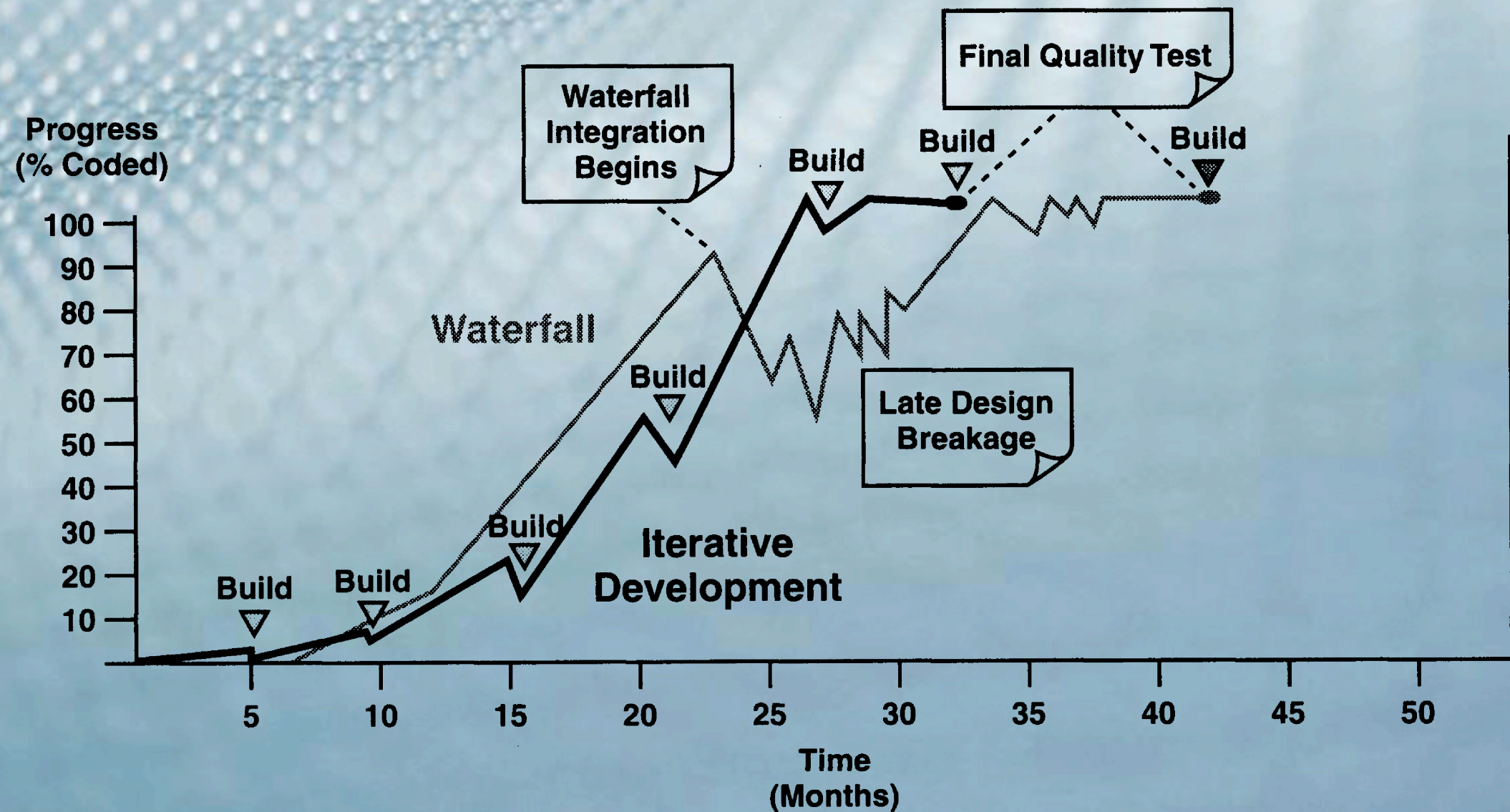
“Schwere Prozesse”

- viele Artefakte, erstellt in einer bürokratischen Atmosphäre
- rigidity und Kontrolle
- Ausführliche, langfristige und detaillierte Planung
- Eher *predictive* als *adaptive*

Leichte, *agile* Prozesse

- iteratives Vorgehen
- Änderungen sind als “Projekttreiber” erwünscht und werden nicht bekämpft
- “Leichter” Prozess, wird kurzfristig den Anforderungen angepaßt
- Artefakte werden in Art und Umfang dem Projekt angepaßt
- Kein detaillierter Plan für das gesamte Projekt
 - High-Level Phase Plan (Milestones)
 - Iteration Plan (plant nur nächste Iteration im Detail)

Vergleich: Wasserfallmodell



Unified Process

Wie funktioniert das?

Die 4 Phasen des Unified Process

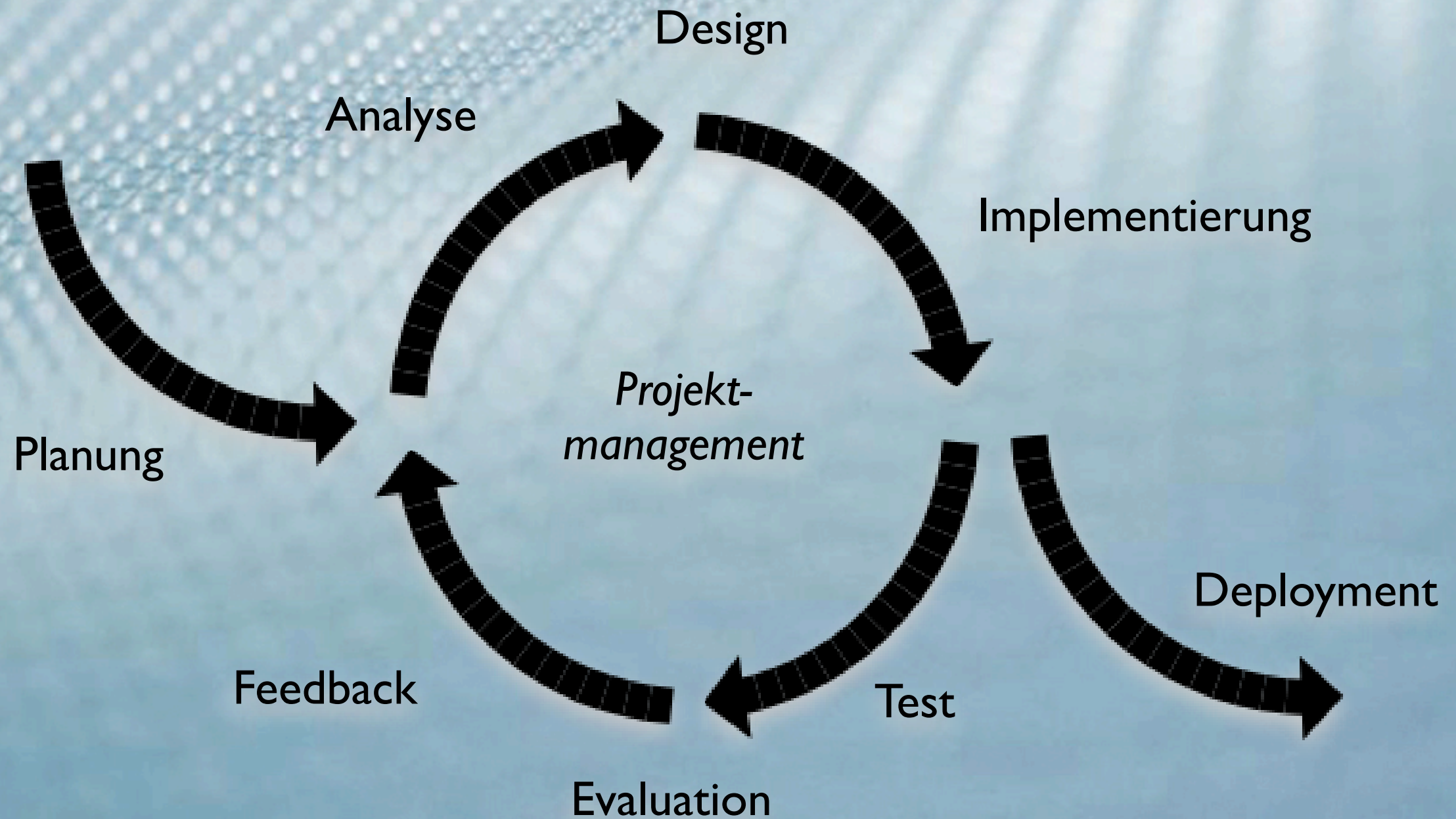
- Beginn (*Inception*)
 - Vision, Kundennutzen,
- Ausarbeitung (*Elaboration*)
 - Implementierung der Architektur und Kernfunktionalitäten
- Konstruktion (*Construction*)
 - Implementierung
- Umsetzung (*Transition*)
 - Abschließende Tests, Inbetriebnahme

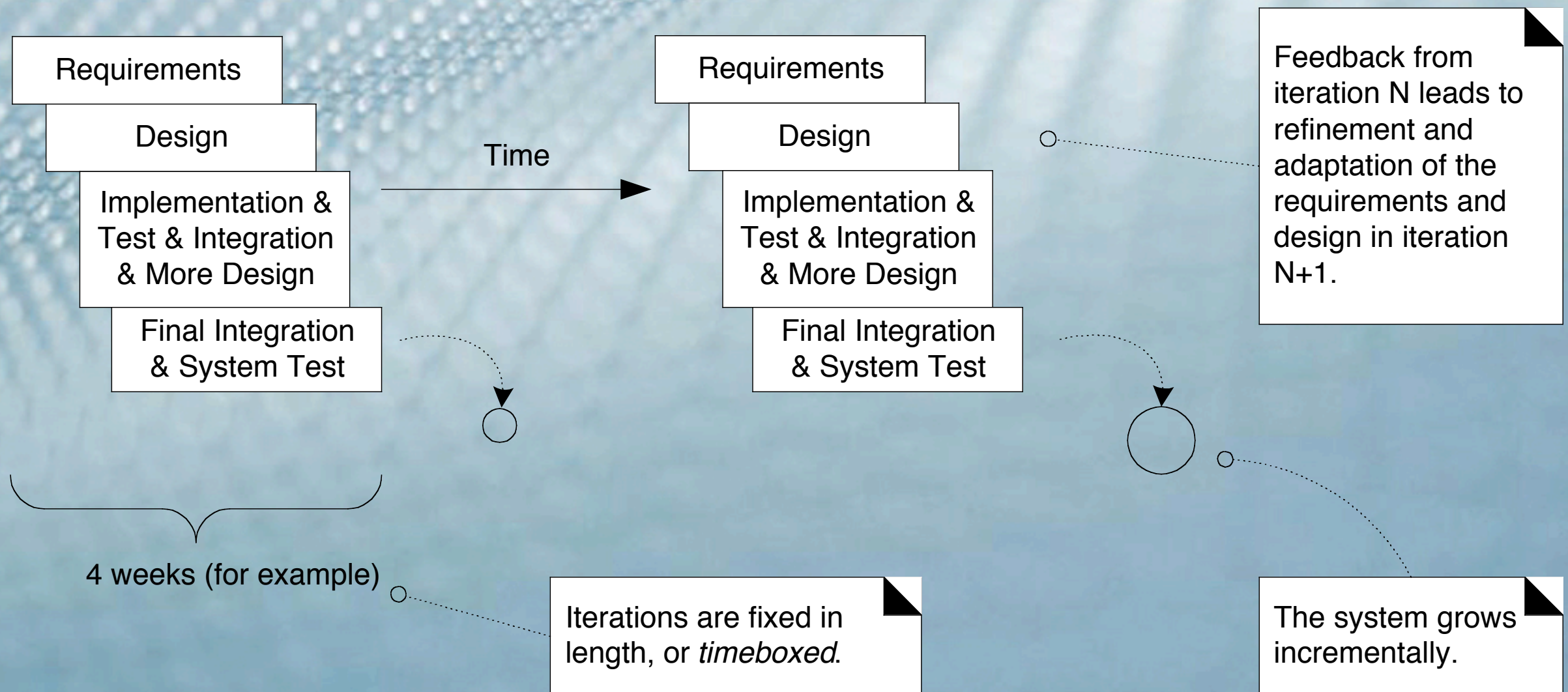
Arbeitsschritte

(Disciplines, Workflows)

- Anforderungen (*business modeling, requirements*)
- Analyse (*analysis, requirements*)
- Entwurf (*design*)
- Implementierung (*implementation*)
- Test
- Inbetriebnahme (*deployment*)
- Projektmanagement
- Qualitätssicherung (*quality assurance*)

Iteration

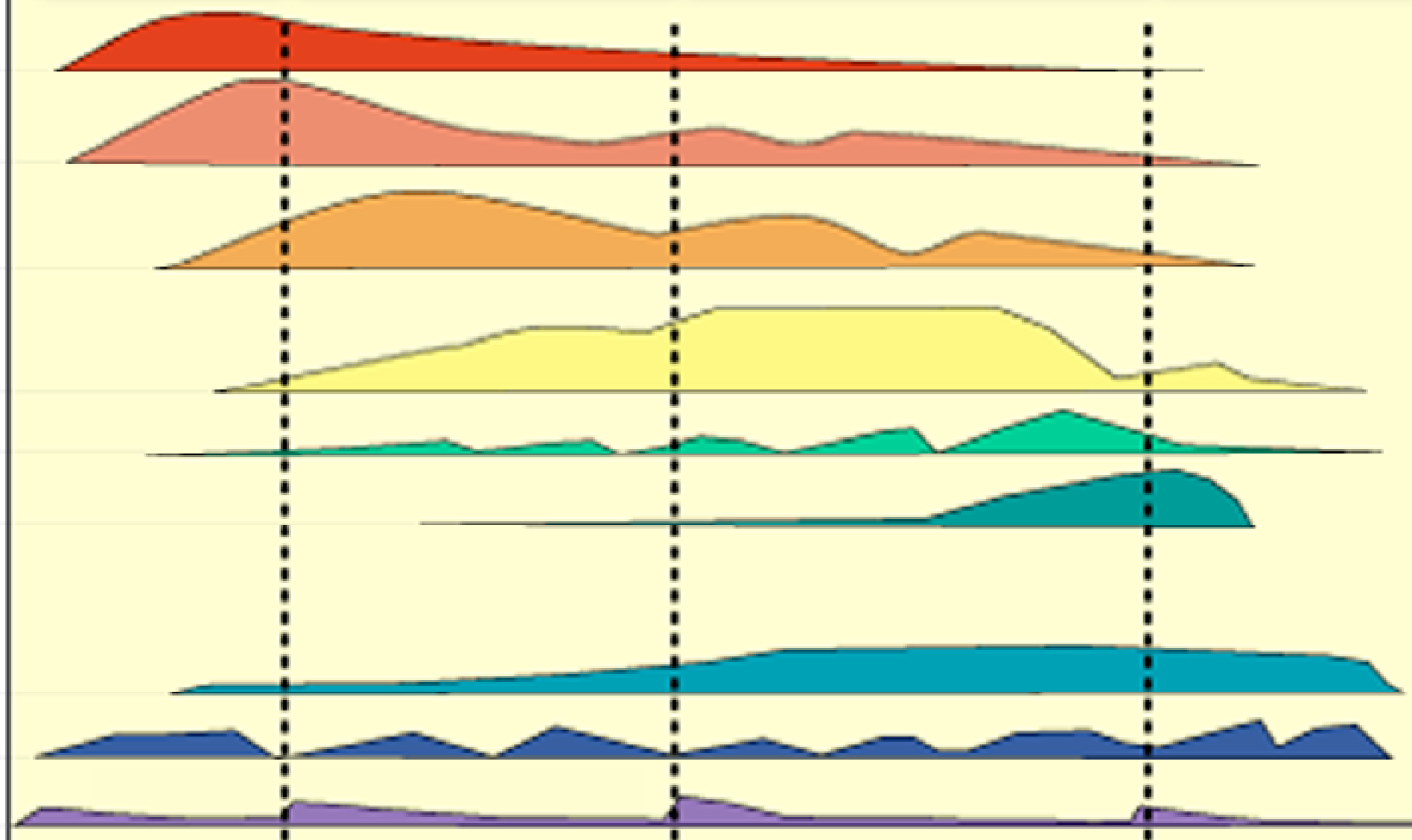




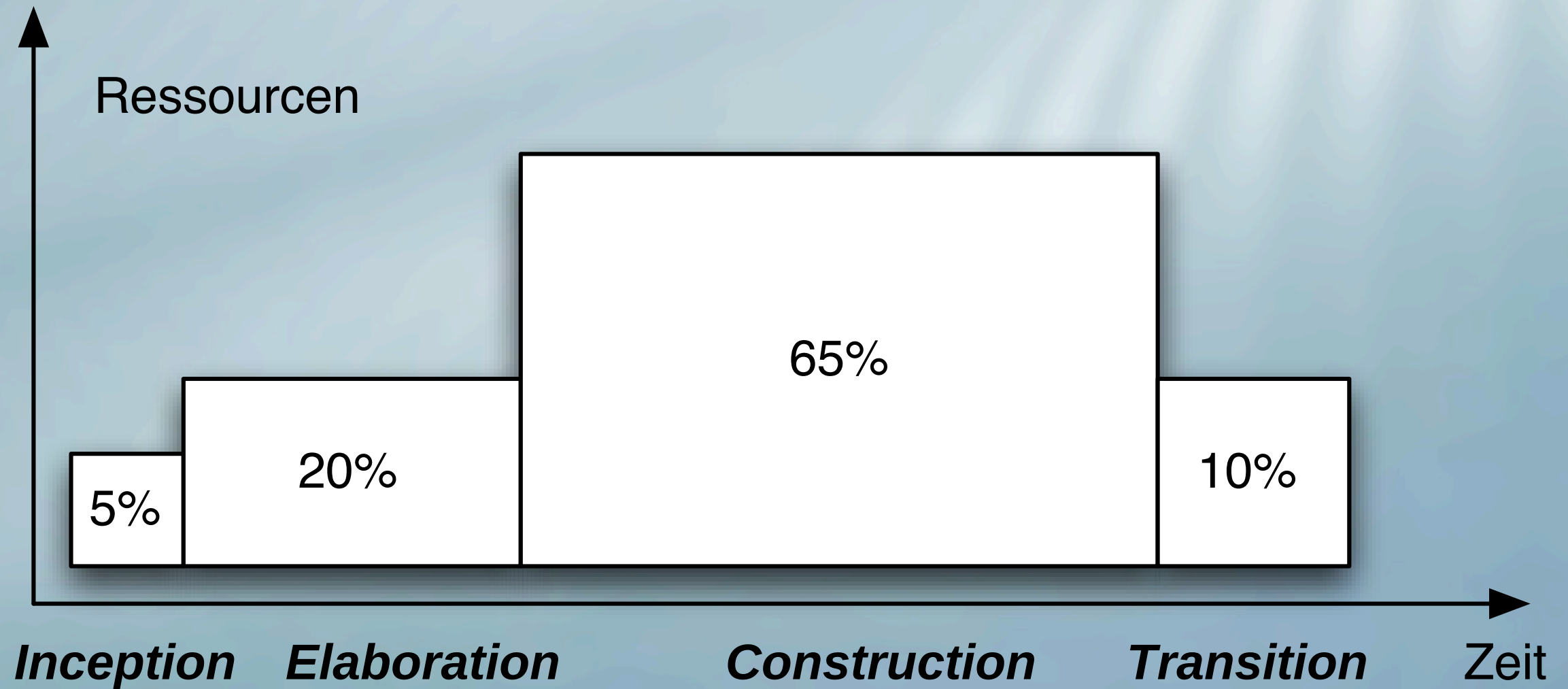
Disciplines

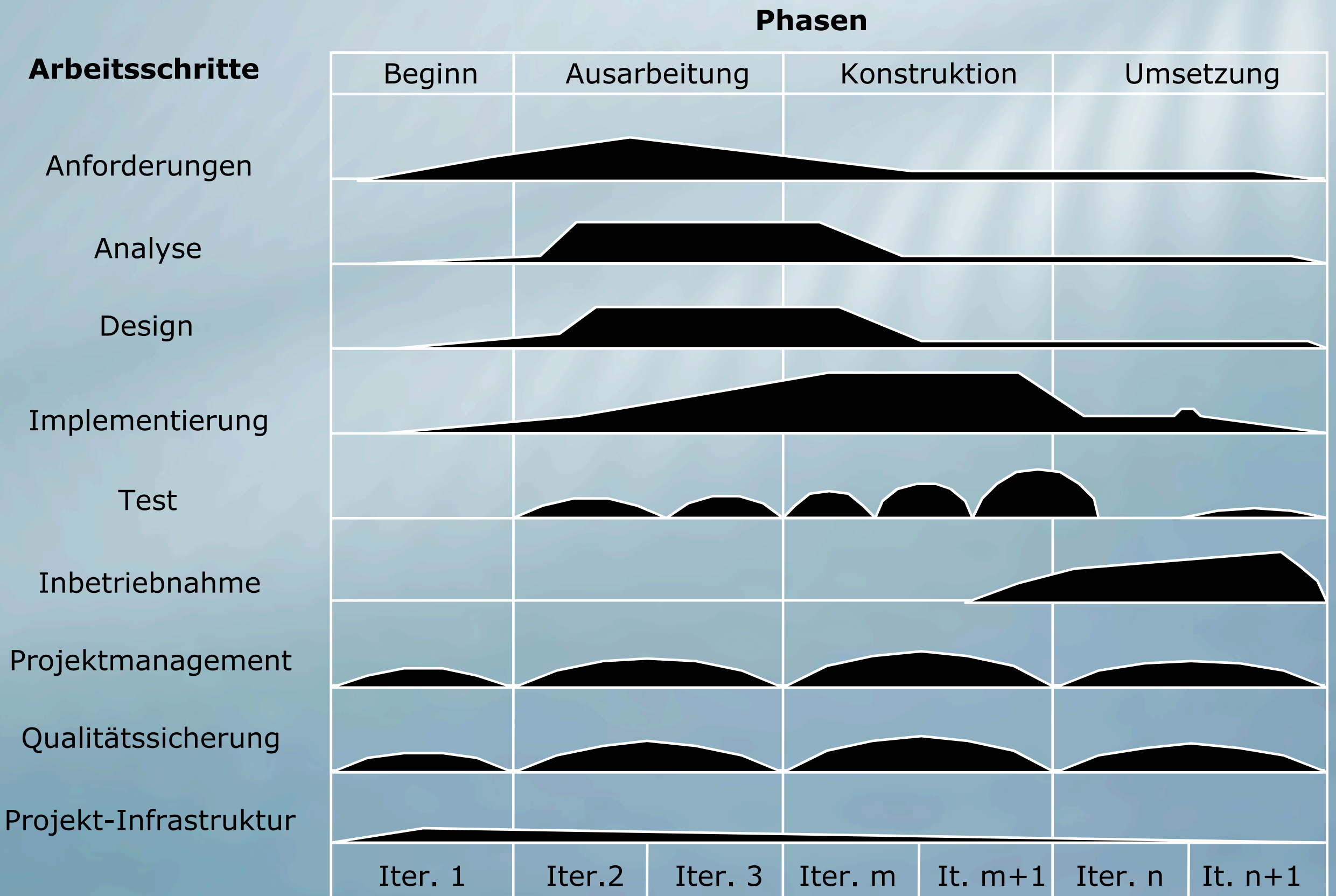
Business Modeling
Requirements
Analysis & Design
Implementation
Test
Deployment
Configuration
& Change Mgmt
Project Management
Environment

Phases



Iterations





Anwendung des Unified Process

- Der Unified Process muss angepasst werden
 - an das spezifische Projekt, das Unternehmen, das Projektteam, etc.
- Einzige Invarianten sind
 - Iterative und Inkrementelle Entwicklung
- Alle Aktivitäten, Artefakte sind *optional*!
 - Fokus auf *wenige* Artefakte, die *hohen praktischen Wert* haben

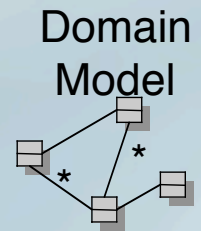
Development Case (Artifact of Environment Discipline)

	Artefakt	Beginn <i>Inception</i>	Ausarbeitung <i>Elaboration</i>	Konstruktion <i>Construction</i>	Umsetzung <i>Transition</i>
Arbeitsschritt		Iteration: I1	E1..En	C1..cn	T1..Tn
Business Modeling	Domain Model		s		
Requirements	Use-Case-Model	s	r		
	Vision	s	r		
	Supplementary Specification	s	r		
	Glossary	s	r		
Design	Design Model		s	r	
	SW Architecture Document		s		
	Data Model		s	r	
Implementation	Implementation Model		s	r	r
Project Management	SW Development Plan	s	r	r	r
Testing	Test Model		s	r	
Enviroment	Development Case	s	r		

s - start; r - refine

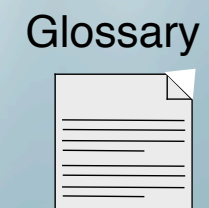
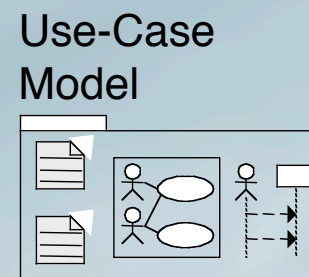
Sample UP Artifacts

Business Modeling

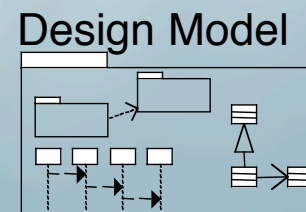


Partial artifacts,
refined in each
iteration.

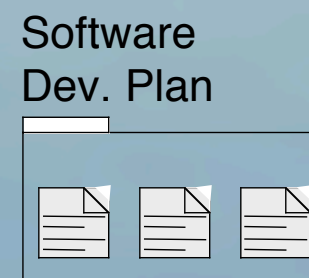
Requirements



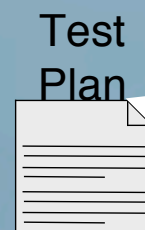
Design



Project Management



Test



Environment

