

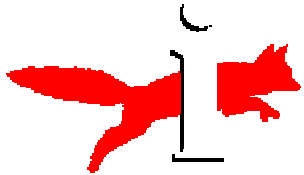
Requirements Engineering

- Was versteht man unter RE?
- Warum ist es wichtig?
- Einbettung in Projekt-Vorgehensmodellen
- Eigenschaften eines Requirements Engineers
- Ausbildungsmöglichkeiten



© 2009 E. Fuchs





Was versteht man unter „Anforderung“ (Requirement)?

□ Anforderung (Requirement):

- Eine **Fähigkeit** oder **Voraussetzung**, **Bedingung**, **Eigenschaft**, die ein **System** (Maschine, Software, Person, etc.) **erfüllen** oder besitzen **muss**, um einen **Vertrag**, eine **Norm**, oder ein anderes formell bestimmtes Dokument **zu erfüllen** (IEEE 610.12-1990)

□ Anforderung (Requirement) kurz:

- Eine Bedingung oder Fähigkeit, die das System erfüllen muss

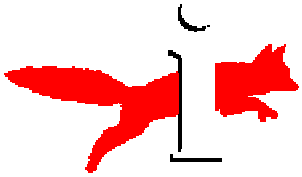
□ Anforderung (Requirement) anders:

- Anforderung ist eine extern beobachtbare Eigenschaft einer gewünschten Entität (Alan Davis, 2005)



Anforderungen:
- Genauigkeit?
- Sportlich?
- Design?

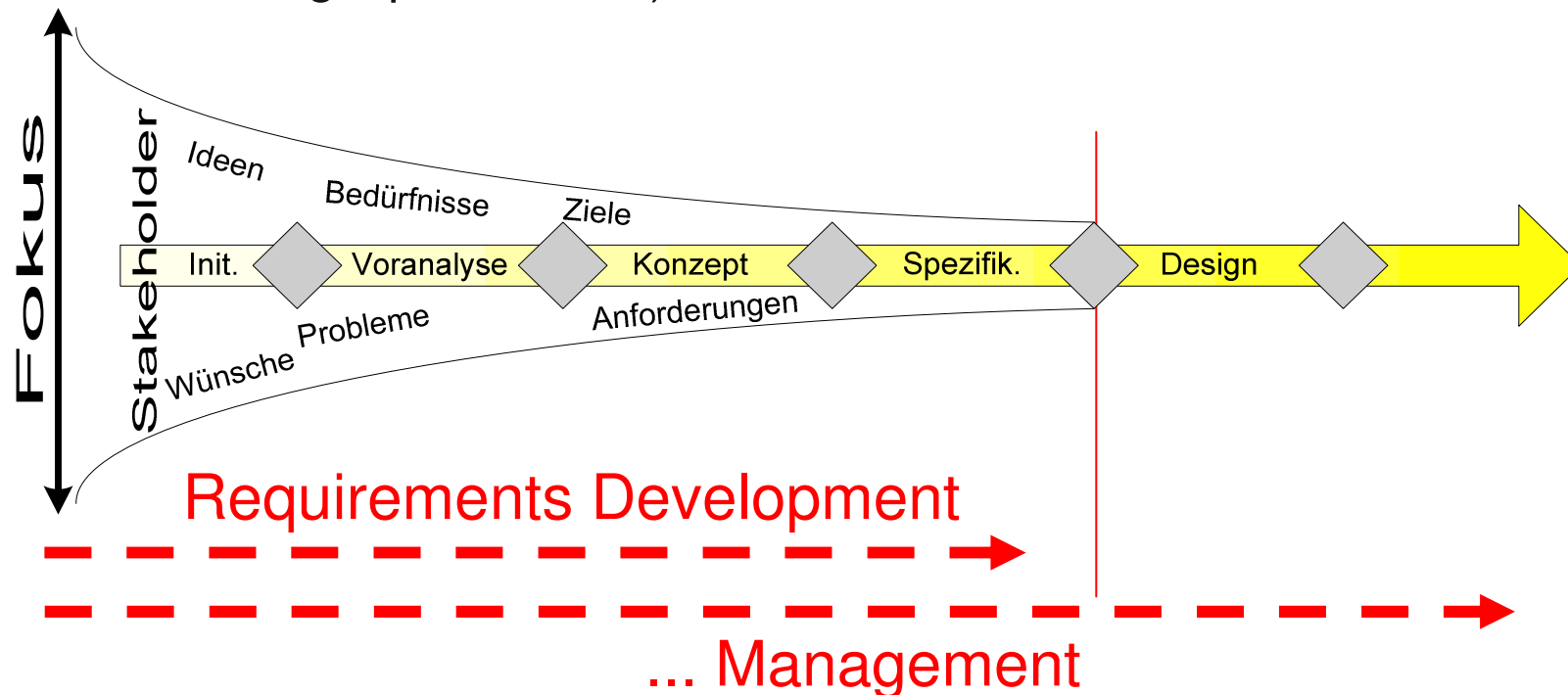


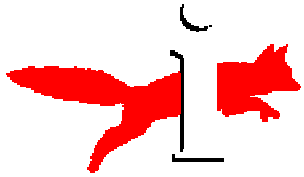


Was versteht man unter RE?

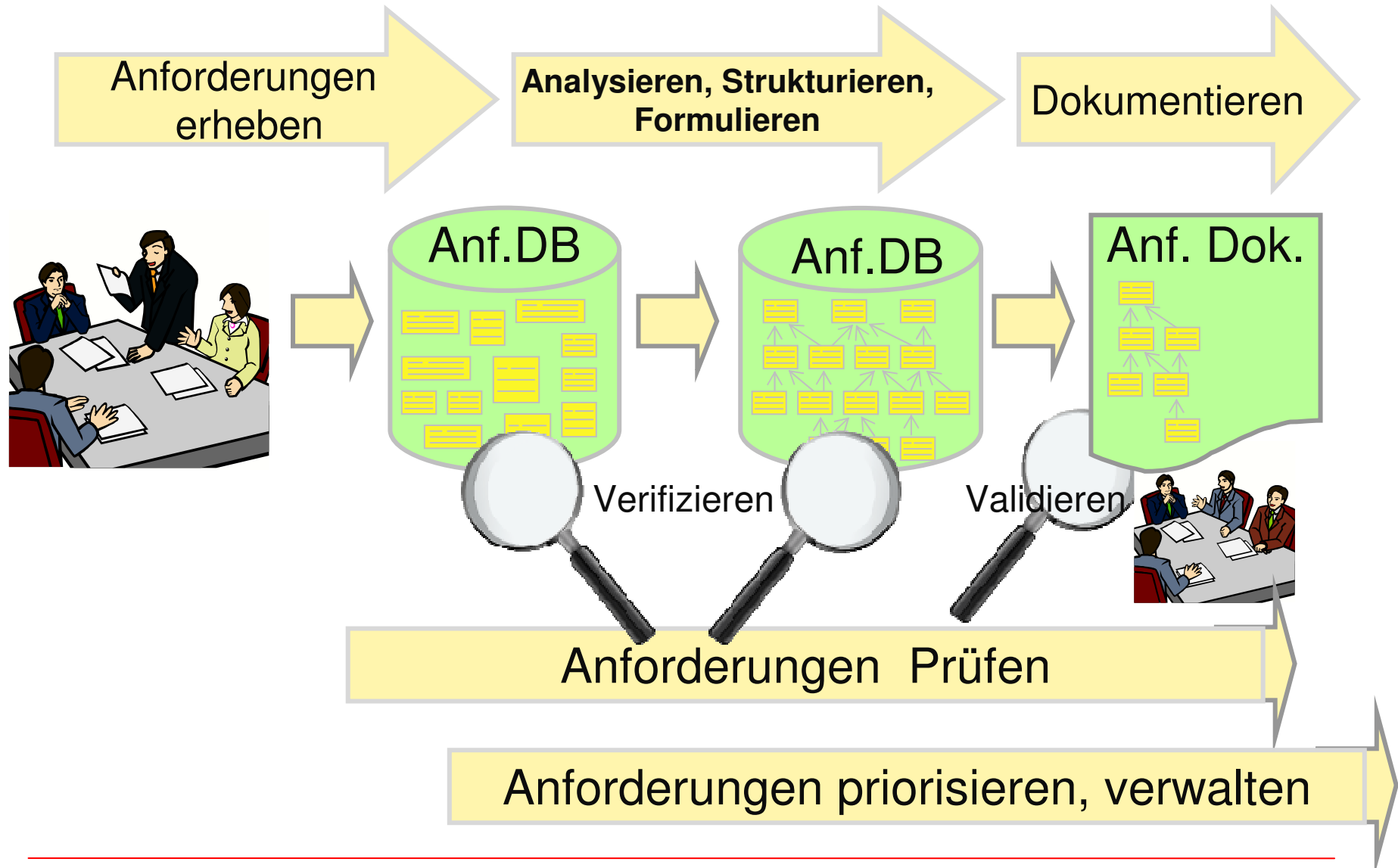
Die systematische Vorgehensweise, um die Anforderungen in einem – oft iterativen – Prozess zu ermitteln, analysieren, priorisieren, spezifizieren und verwalten bezeichnet man als **Requirements Engineering (RE)**

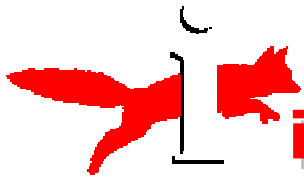
Synonyme: (Systemanalyse / Anforderungsanalyse / Anforderungsspezifikation).



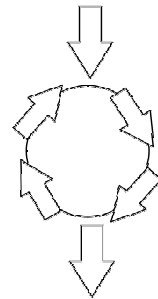
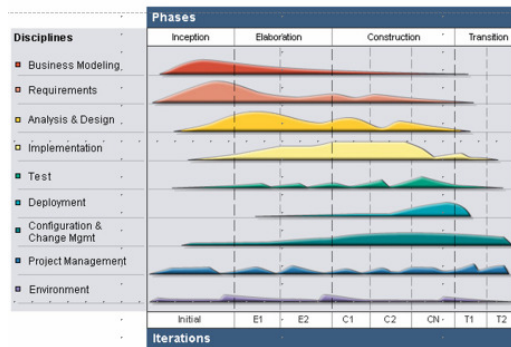


Requirements Engineering: die Haupttätigkeiten

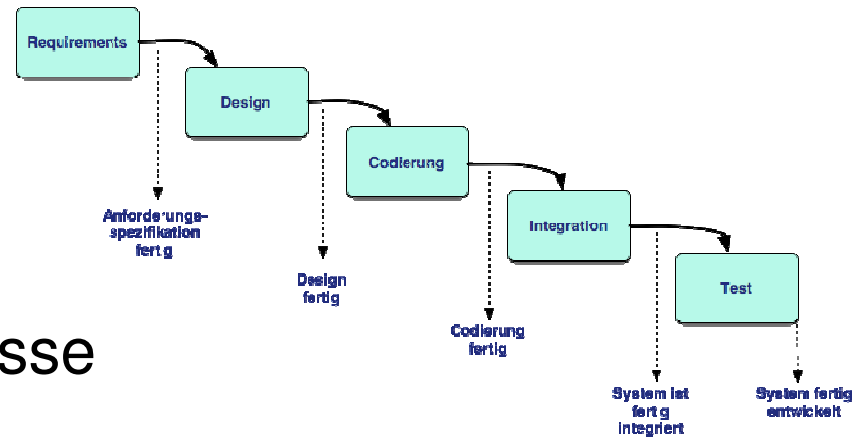




Requirements Engineering ist abhängig von Organisation und Prozess



Prozesse

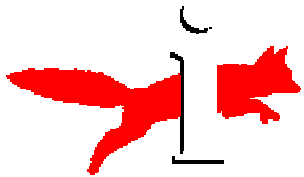


Best practices

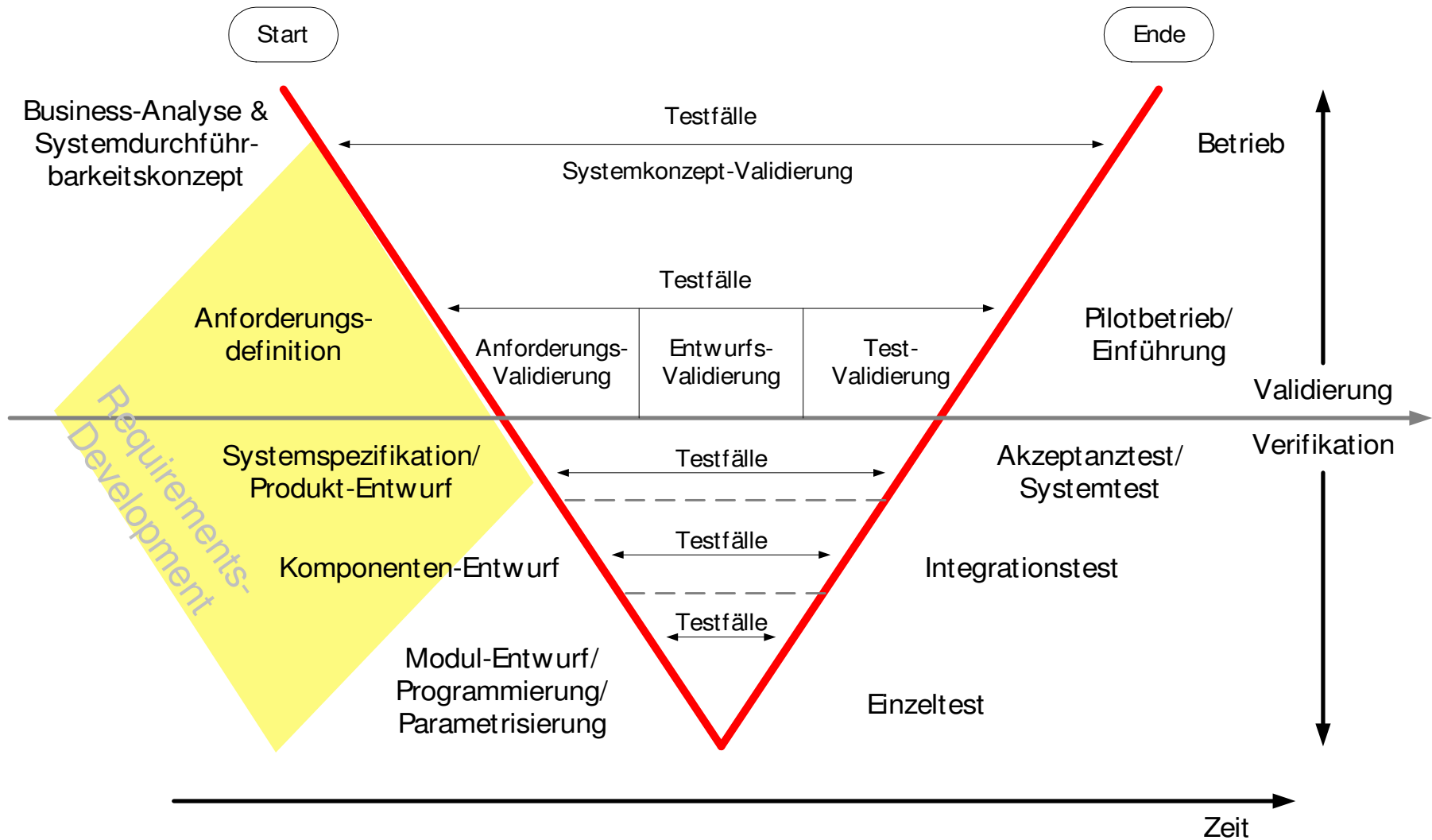


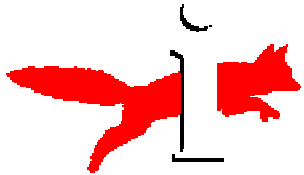
Normen und Standards



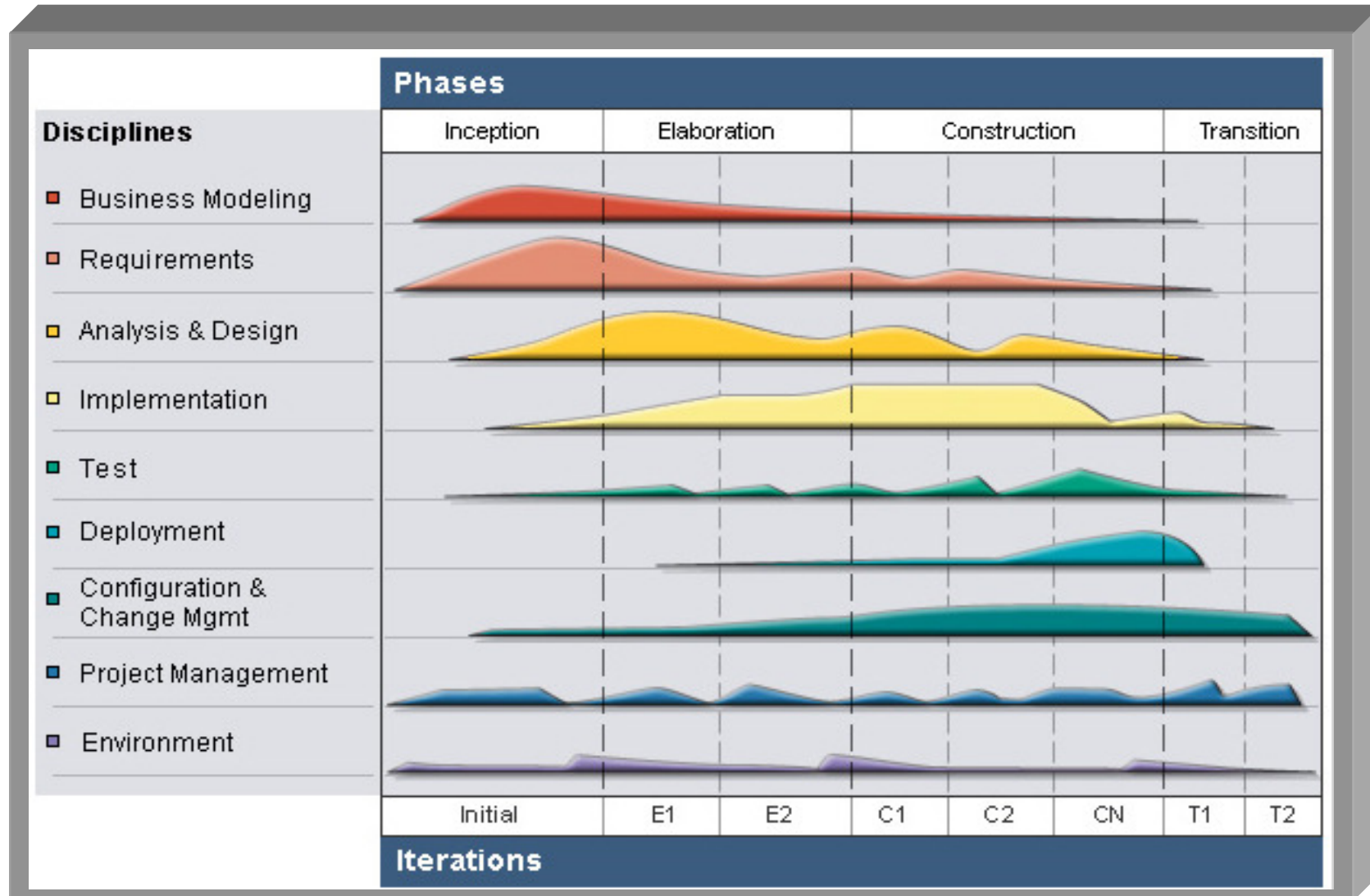


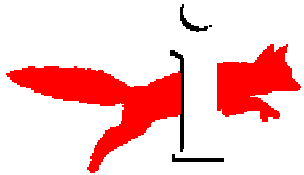
RE und Projektvorgehen: im V-Model





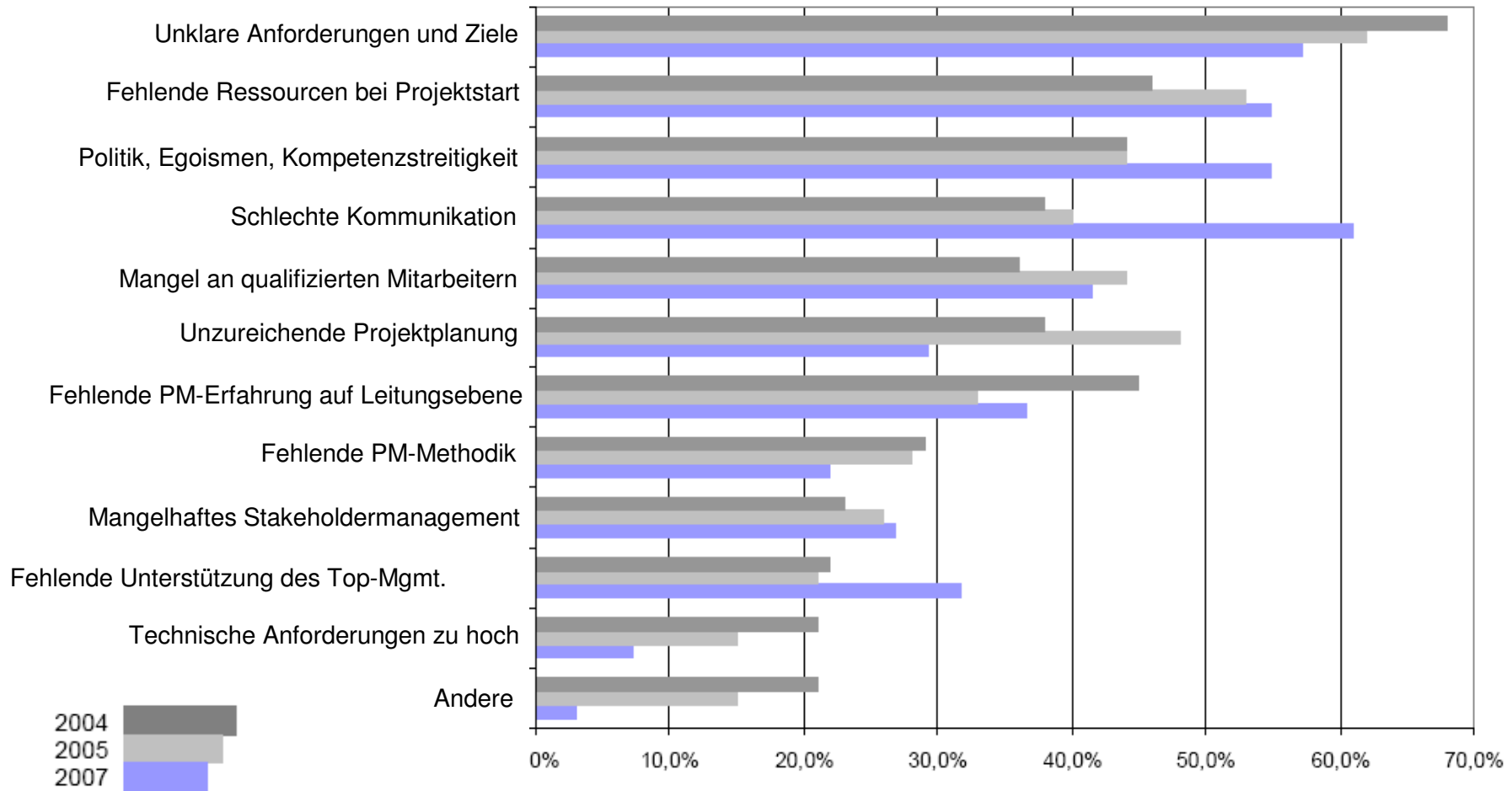
RE und Projektvorgehen: im RUP





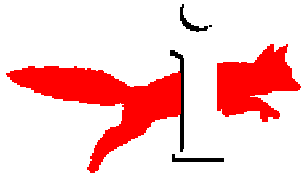
Motivation für das RE: Warum Projekte scheitern

□ Studie zu Misserfolgsfaktoren von Projekten



Quelle: GPM (DE), PA Consulting 2007

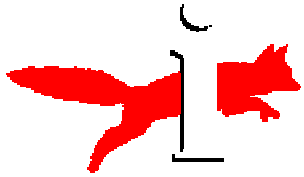




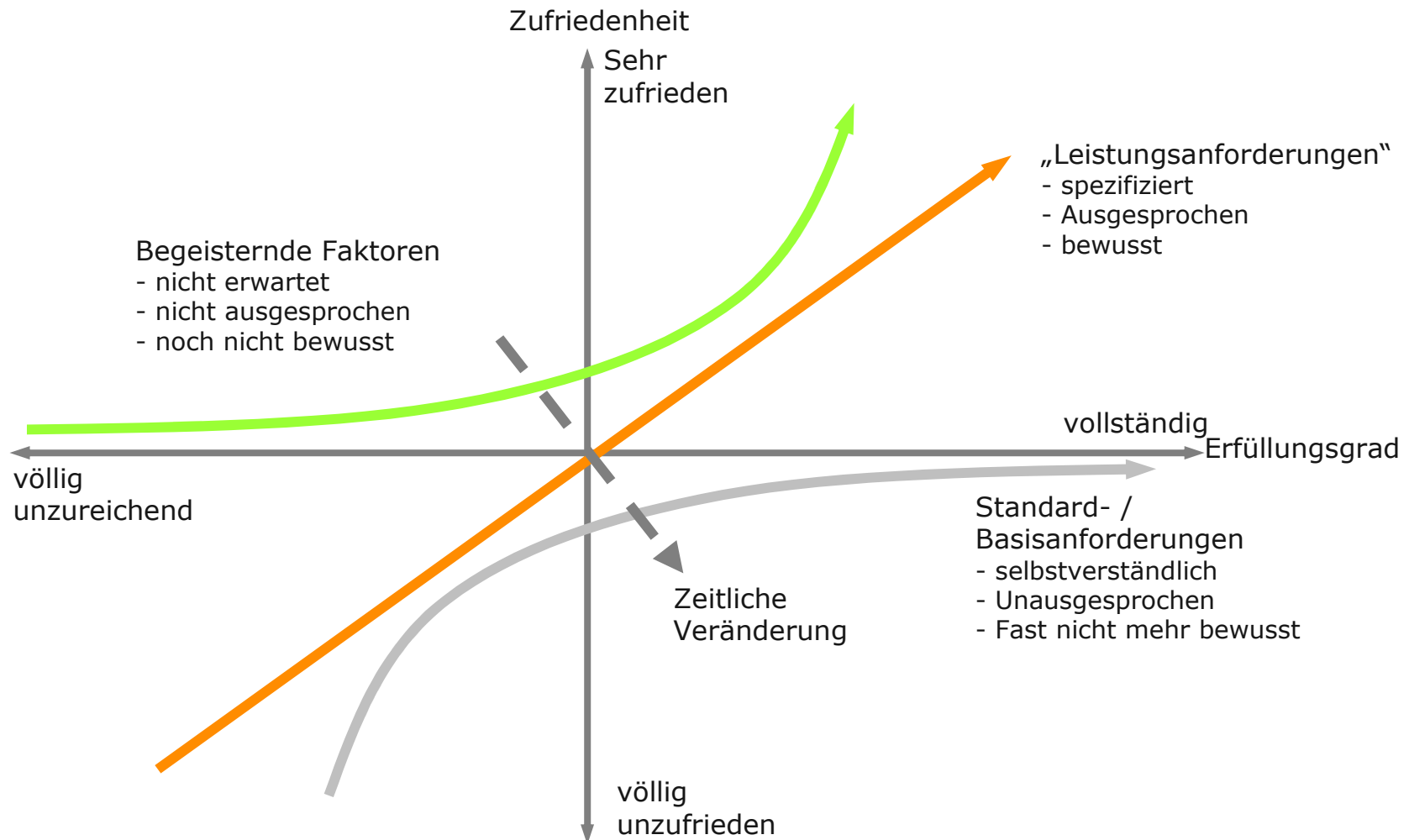
Warum ist RE so schwierig ???

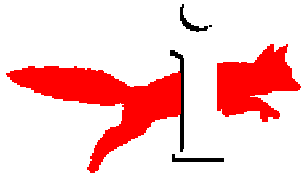
- ❑ Viele Stakeholder mit unterschiedlichen Zielen, Bedürfnissen
- ❑ Vielseitige Geschäftsprozesse
- ❑ Unterschiedliche Arbeitsweisen
- ❑ Kommunikationsprobleme
- ❑ Undurchsichtiges, komplexes Ist-System
- ❑ Immaterialität (Software, Daten)
- ❑ Vielseitige Möglichkeiten für das Soll-System
- ❑ Schnell ändernde Arbeitswelt
- ❑ Oft zu wenig Fach- und IT-Wissen vorhanden
- ❑ Termin- und Budgetdruck (schnelle Ergebnisse)





Kategorien von Anforderungen: Das Kano-Modell





Nutzen des Requirements Engineerings

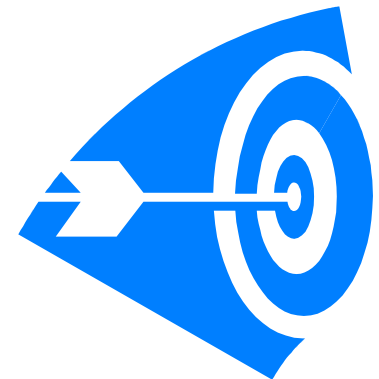
- ❑ Man entwickelt die „richtige“, d.h. die Benutzer zufriedenstellende Lösung schneller, einfacher und billiger

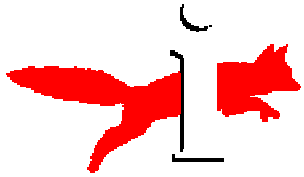
- Weniger Abgrenzungsprobleme des Systemumfanges
- Weniger Missverständnissen
- Weniger Fehler in der Spezifikation
- Weniger Änderungsanträge während der Entwicklung
- Weniger, aber zielgerichtete Features
- Weniger Kosten einer Weiterentwicklung

- Weniger Chaos im Projekt
- Bessere Schätzungen
- Bessere Ausgangsposition für das Testen

- ❑ Man erreicht

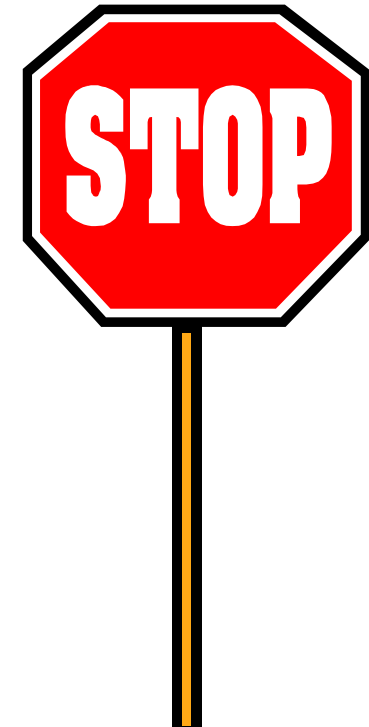
- Höhere Zufriedenheit im Team und beim Kunden / Auftraggeber
- Die Projektziele!

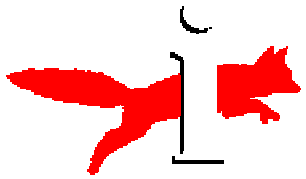




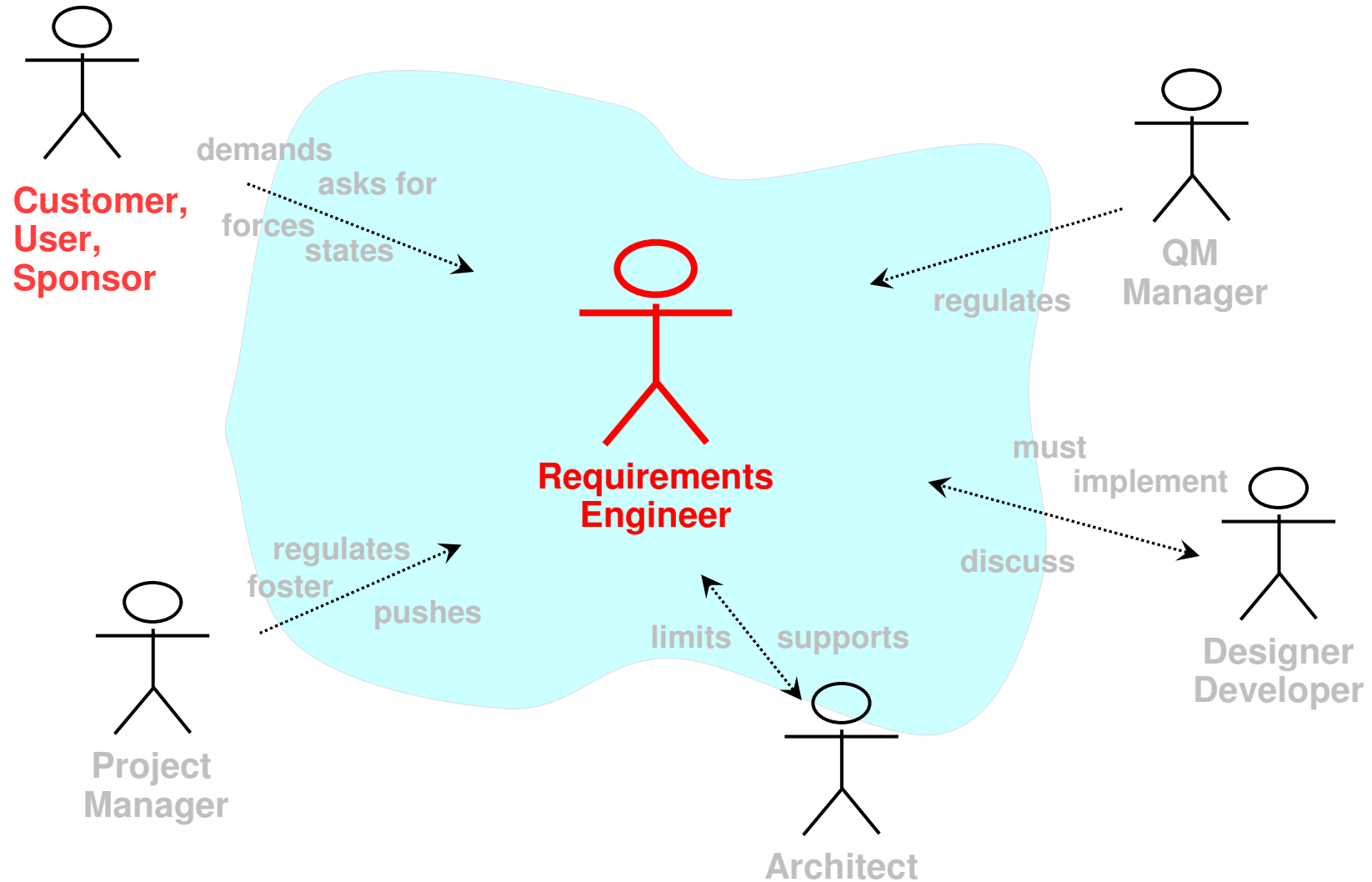
Abgrenzung des Requirements Engineering

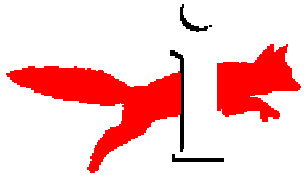
- ❑ Requirements Engineering ist **NICHT**...
 - ❑ System Design
 - Architektur
 - Testing
- Anforderungen sind **NICHT**...
- So viele Dokumente wie möglich
 - Ein möglichst grosses Dokument
 - Ein möglichst perfektes Dokument
 - Ein Roman oder literarisches Werk
 - Eine wissenschaftliche Arbeit



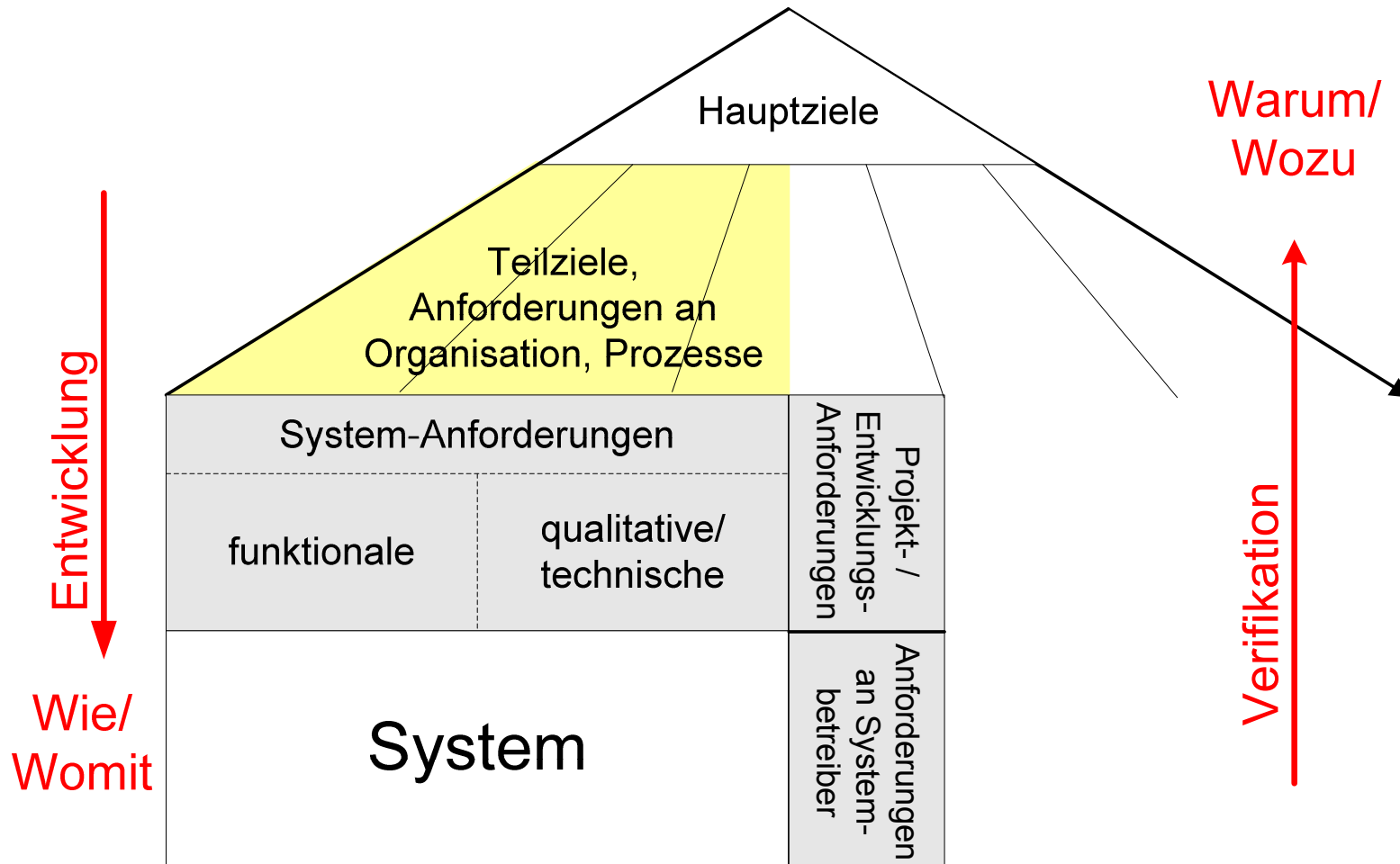


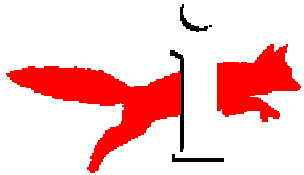
Arbeitsumfeld des Requirements Engineer



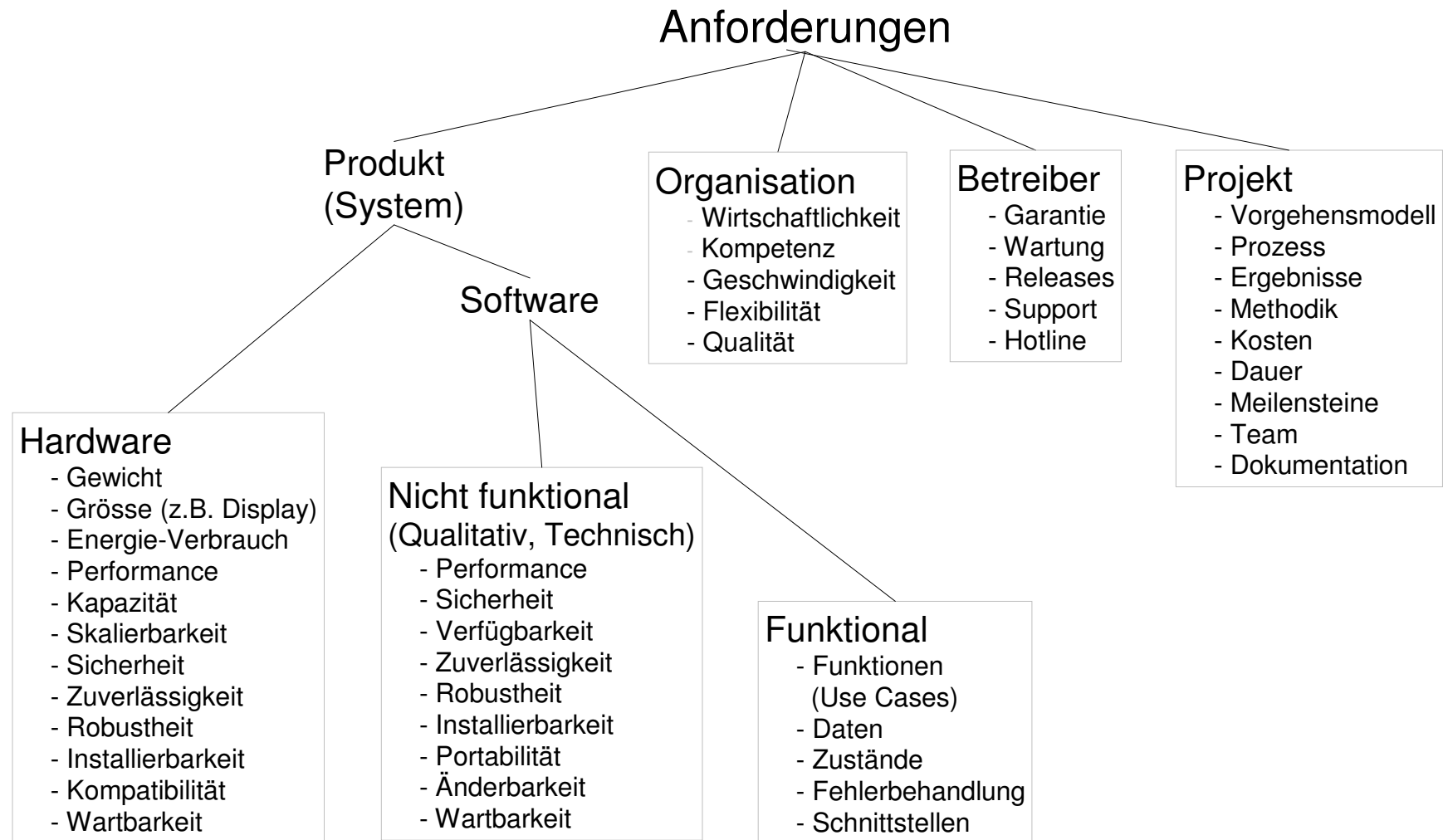


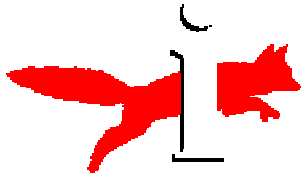
Einordnung und Arten von Anforderungen





Anforderungsarten: Gliederung hilft bei der Erhebung und Prüfung

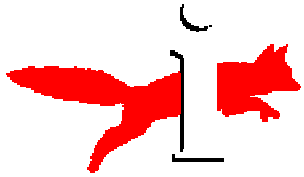




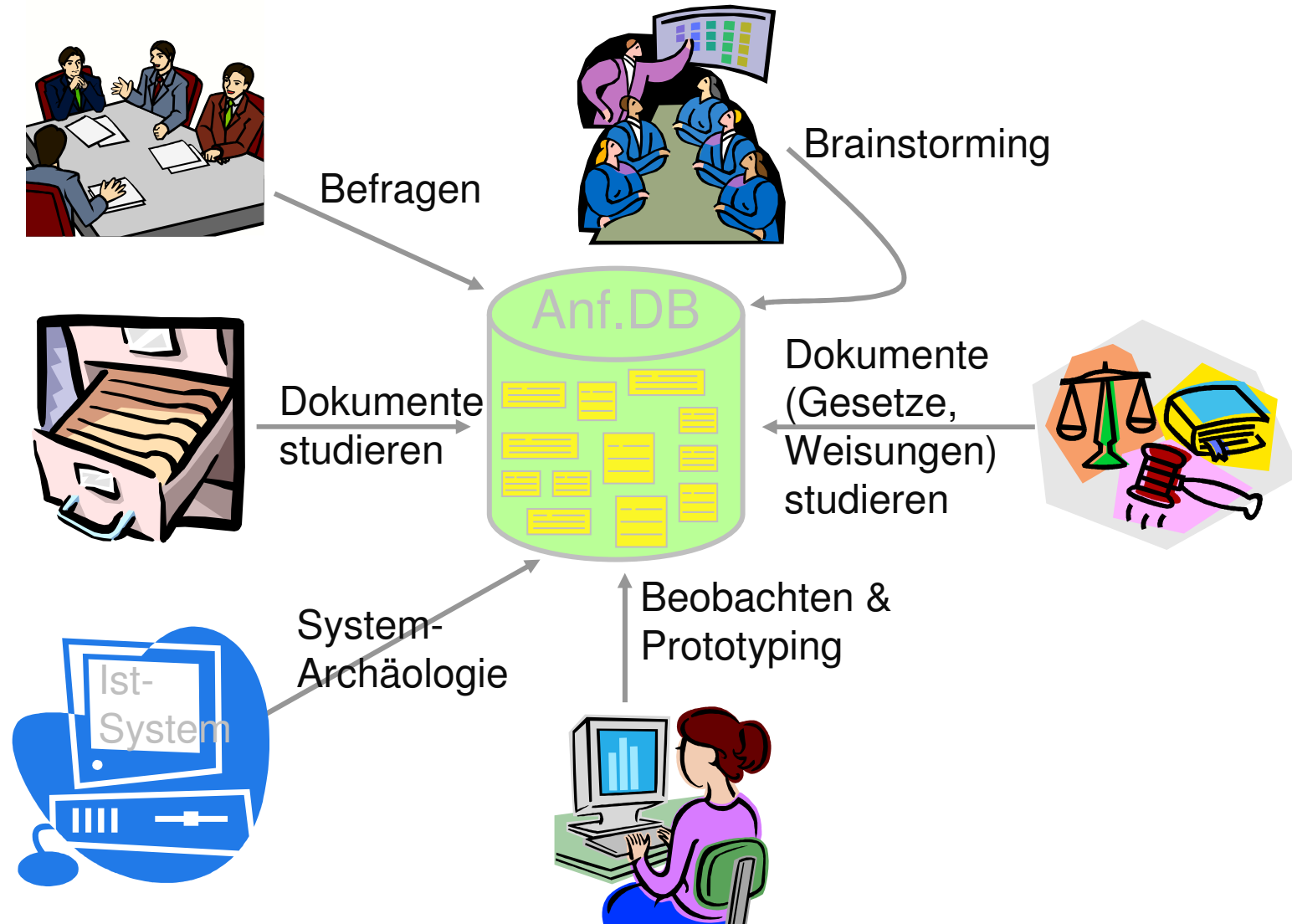
Beschreibung der Anforderungen

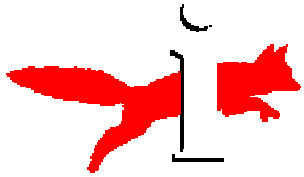
Attribute	
Identifikations-Nr.	Stakeholder, Eigentümer
Bezeichnung	Autor
Kurzbeschreibung	Version
Begründung	Status
Gewicht, Priorität	Prüf- (Review-)Datum
Spezifikation	Weitere Quellen
Risiken, Aufwand	





Ermittlungstechniken

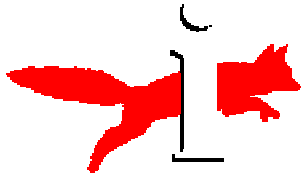




Qualitätskriterien für einzelne Anforderungen

- Abgestimmt
- Bewertet
- Eindeutig
- Gültig und aktuell
- Korrekt
- Konsistent
- Prüfbar
- Realisierbar
- Verfolgbar
- Vollständig
- Verständlich





Formen der Dokumentation

Natürliche Sprache

- textuelle Beschreibungen

Halbformal

- z.B. Use Cases, Entscheidungstabellen
Grafische Modelle

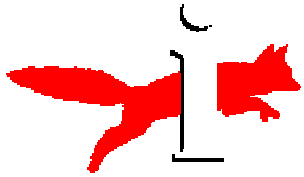
Formal / Algebraisch

- z.B. Algorithmen

flexibler

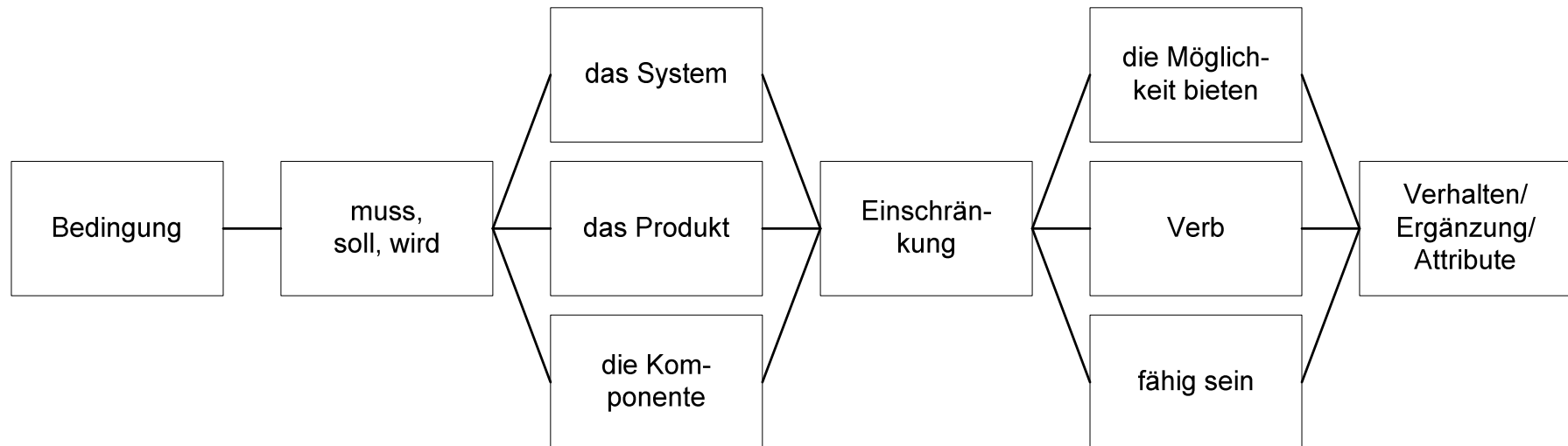
exakter





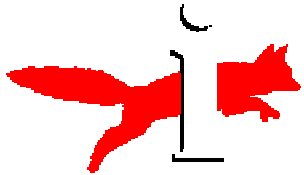
Konstruktion von Anforderungen mittels Satzschablone

- Definition: Eine Satzschablone (Requirements Template) ist ein Bauplan für die syntaktische Struktur einer einzelnen Anforderung.



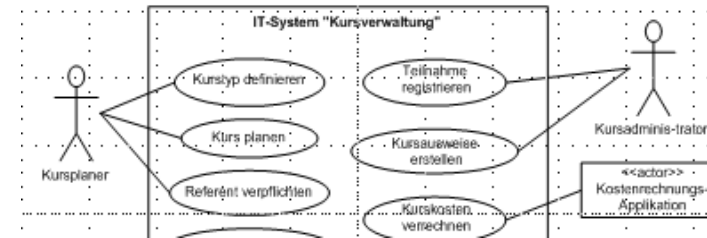
Quelle: Systematisches Requirements Engineering und Management:
Anforderungen ermitteln, spezifizieren, analysieren und verwalten



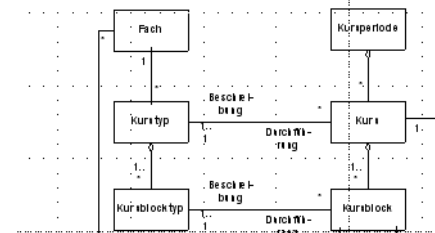


Dokumentation durch konzeptuelle Modelle

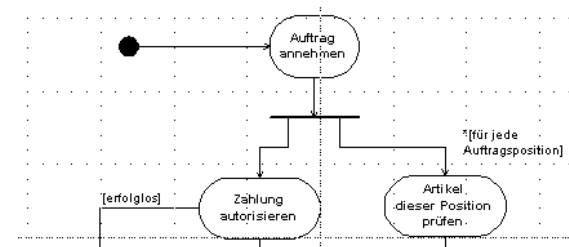
- Überblick über Systemfunktionalität
 - Use-Case-Diagramme



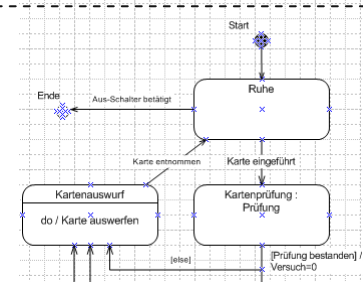
- Begriffssysteme und Datenmodellierung
 - Klassendiagramme

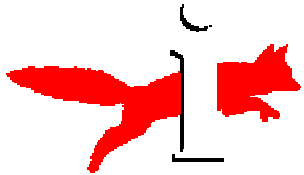


- Ablaufmodellierung
 - Aktivitätsdiagramme



- Ereignisgesteuertes Verhalten
 - Zustandsdiagramme

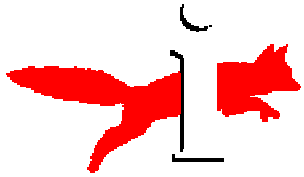




Anforderungsdokument: Beispiel

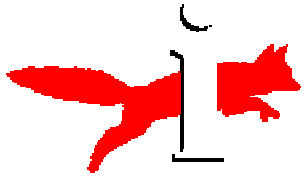
1. Einleitung
Zweck, Leserkreis, Referenzen, Glossar, Konventionen
2. Übersicht
Kontext, Benutzergruppen, Konzepte, Einschränkungen, Annahmen
3. Anforderung an die Organisation, Prozesse
4. Anforderungen an Schnittstellen
5. Anforderungen an Datenhaltung
6. Betriebsarten
7. Funktionale Anforderungen
8. Abhängigkeiten unter den Funktionen
9. Leistungsanforderungen und Mengengerüst
10. Andere Merkmale
11. Konfigurierung, Inbetriebnahme und Betrieb
12. Diverses





Verwendung von Anforderungsdokumenten

- für die Planung
- für den Architekturentwurf
- für die Implementierung
- für den Test
- beim Änderungsmanagement
- bei der Systemnutzung und Systemwartung
- im Vertragsmanagement



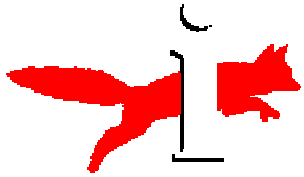
Techniken zur Prüfung von Anforderungen

- ❑ Statische Techniken
 - Reviews (Stellungnahme, Inspektion, Walkthrough)
- ❑ Perspektivenbasiertes Lesen
- ❑ Einsatz von Checklisten
- ❑ Dynamische Techniken
 - Prototypen (Verhalten, Design, Usability)
 - Simulation (Verhalten)

- ❑ Abgleich verschiedener Darstellungsformen
 - Prüfung von Analyse-Modellen mit Werkzeugen (RE-, UML-Tools, ...)
 - durch Menschen
 - Storyboards gegen Use Cases
 - Use Cases gegen Datenmodell
 - Use Cases gegen Zustandsdiagramm, u.a.

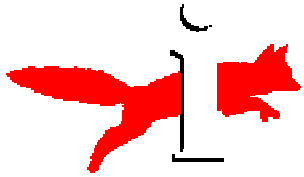
Die Kombination der Vorgehensweisen führt zu den besten Ergebnissen





Anforderungen verwalten

- ❑ Ziel: Transparenz für Kunde, Projektteam und Management
 - Projektfortschritt kontrollieren
 - Entwicklung planen
 - Änderungen verwalten



Konfiguration

Gruppierung von
Anforderungsversionen:
- sachlogisch zusammenhängend
- konsistent
- nicht veränderbar

eindeutiger
Identifikator

Anforderungskonfiguration 001

Anforderung 001

v0.1

v0.2

v0.3

Anforderung 002

v0.1

v0.2

v1.0

v1.1

Anforderung 003

v0.1

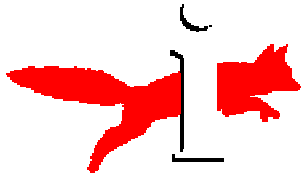
Anforderung 004

v0.1

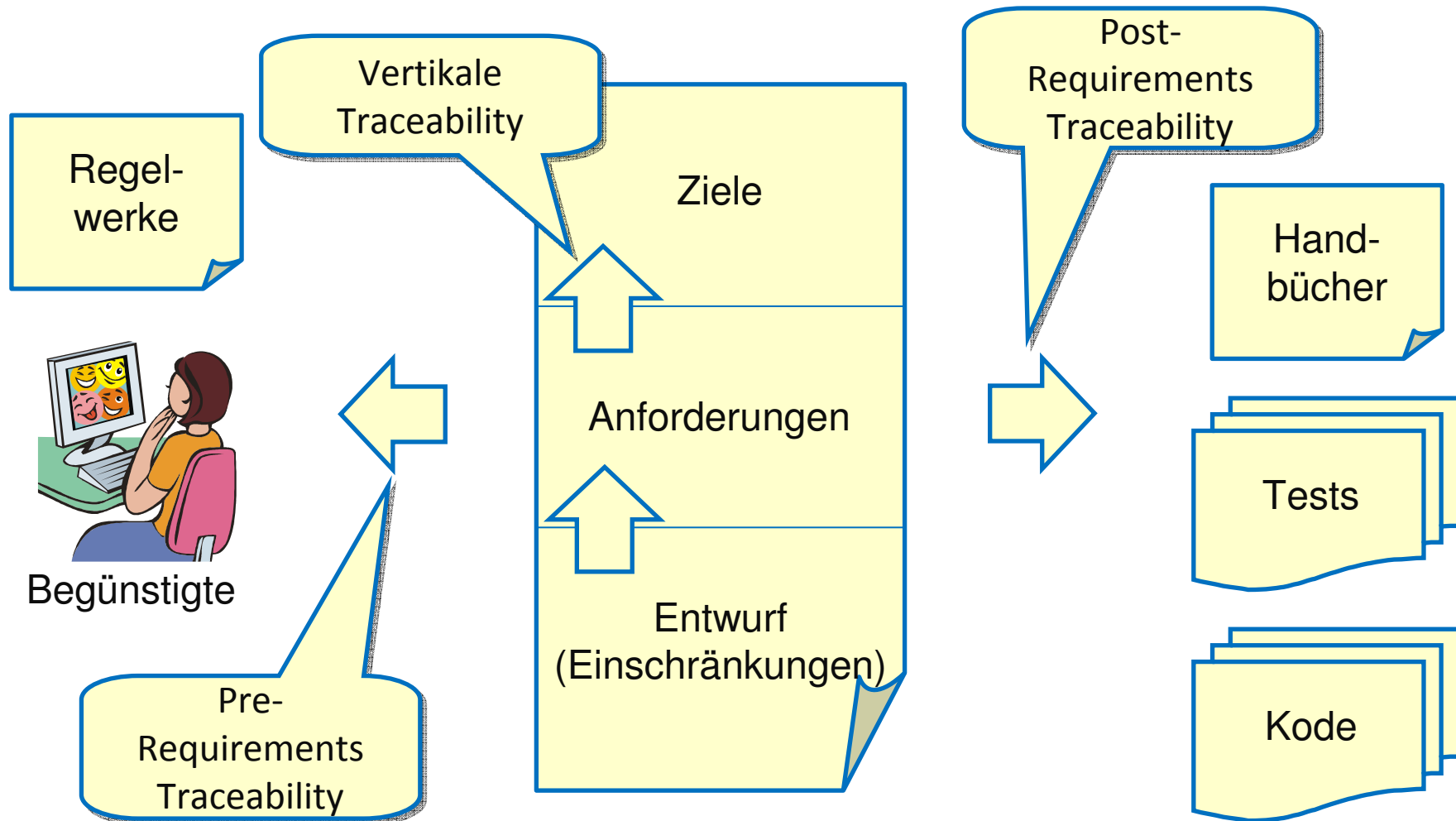
v0.2

Anforderung 005

v0.1

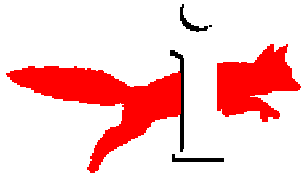


Nachverfolgbarkeitsbeziehungen

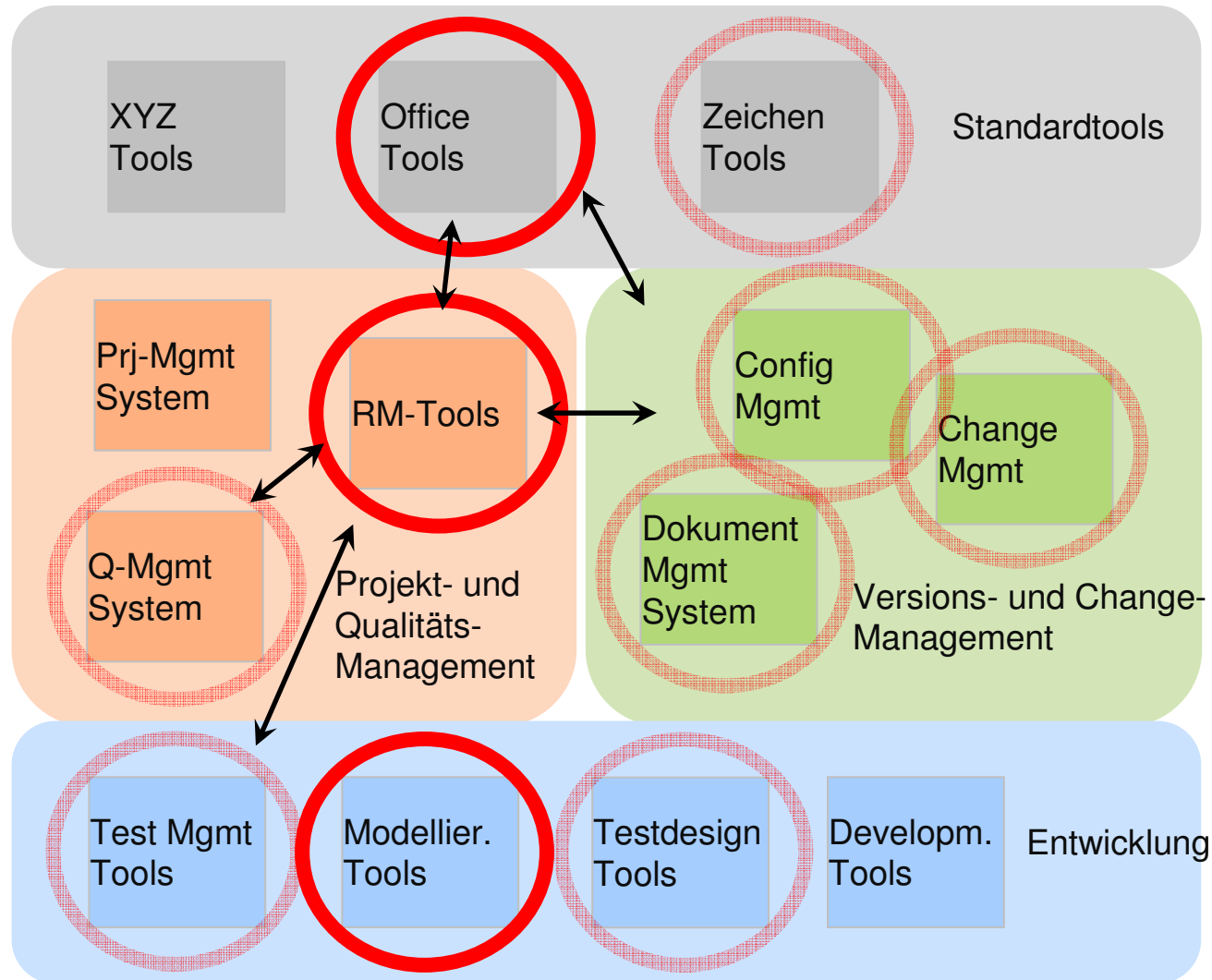


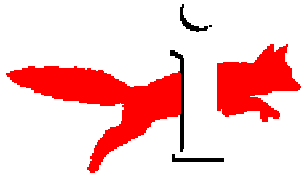
Gotel, Finkelstein (1994): "An Analysis of the Requirements Traceability Problem". ICRE'94.
Gorschek, Wohlin (2006): "Requirements Abstraction Model". Requirements Engineering.



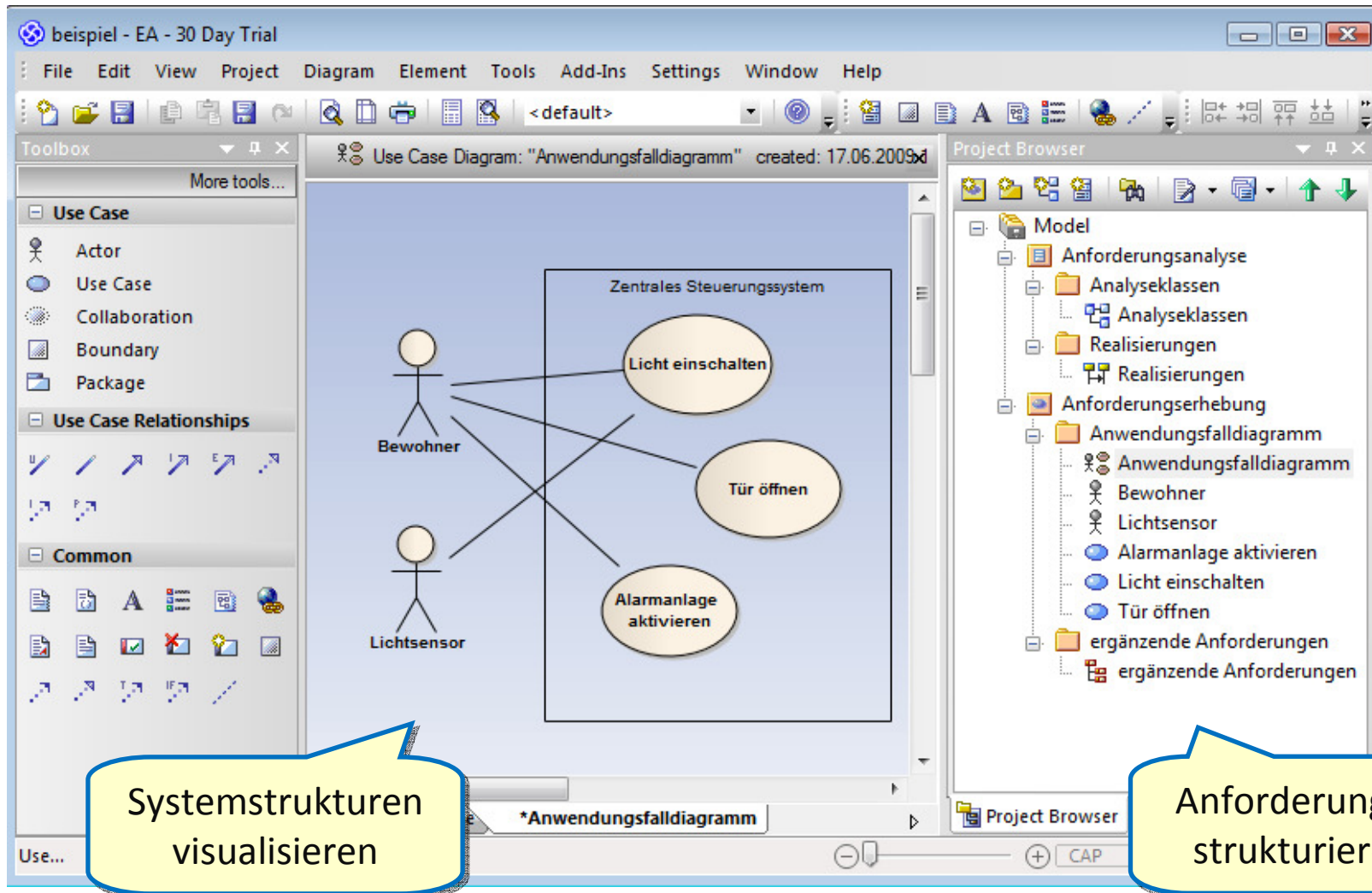


Einsatzzweck von Werkzeugen



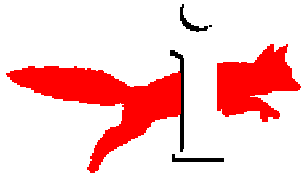


Beispiel: SparxSystems EnterpriseArchitect



Gerber (2009): "Anforderungsmodellierung mit UML und UP mit EnterpriseArchitect von SparxSystems". Seminararbeit Universität Zürich.

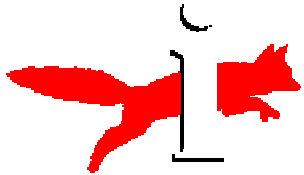




Eigenschaften eines Requirements-Engineer

- Analytisches Denken
- Empathie
- Kommunikationsfähigkeit
- Konfliktlösungsfähigkeit
- Moderationsfähigkeit
- Selbstbewusstsein
- Überzeugungsfähigkeit

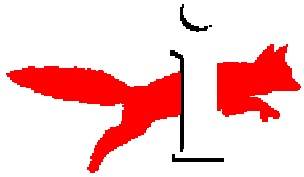




Ausbildungsmöglichkeiten

- ❑ Uni Zürich (Studium)
- ❑ Fachhochschulen (Weiterbildung CAS, MAS)
 - St. Gallen
 - Bern
 - Rapperswil
 - Basel ab 2010)
- ❑ Firmen (Kurse)
 - FUCHS-INFORMATIK AG
 - IFA
 - INFOGEM AG
 - Synspace
 - Zühlke





Certified Professional for Requirements Engineering, Foundation Level

□ Beteiligte Personen und Organisationen

