
Scrum

Agiles Projektmanagement Ansätze für die Praxis



„Scrumtinizing“

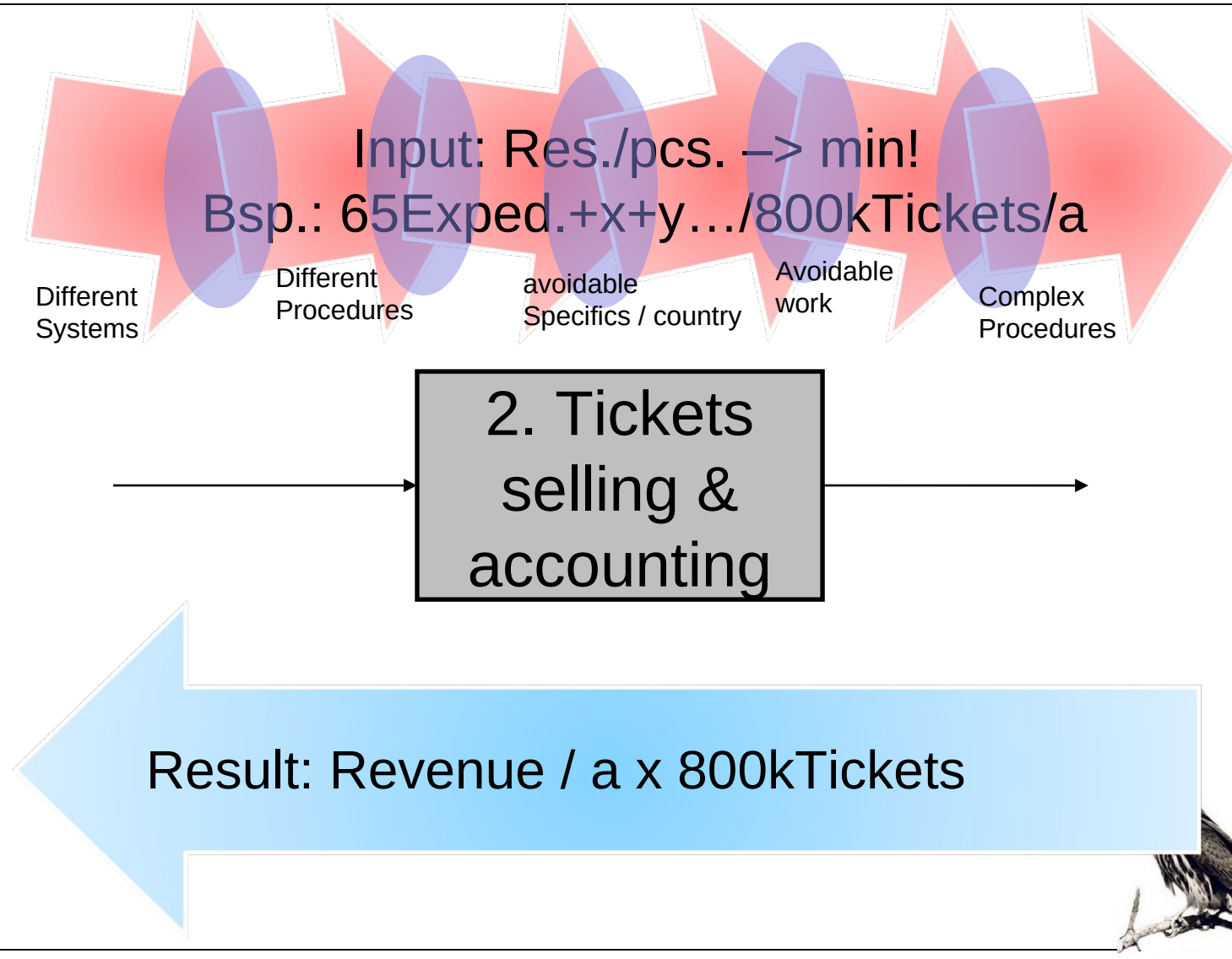
- Worum geht's in der Praxis?
- Warum wurden welche Ideen (nicht!) genommen?
- Gibt's wesentliche Erkenntnisse / Verbesserungen?



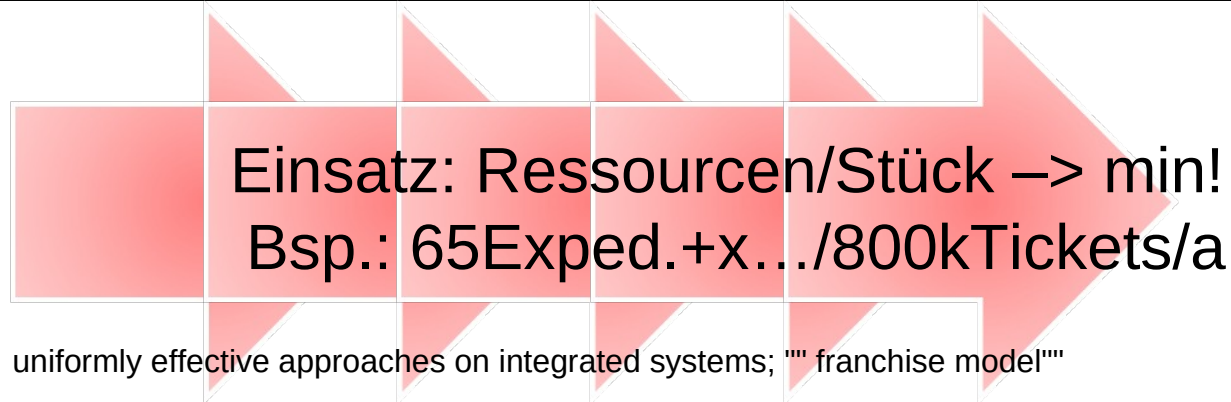
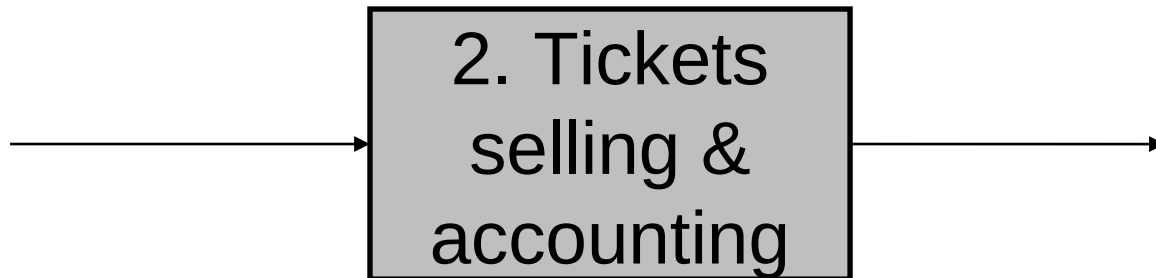
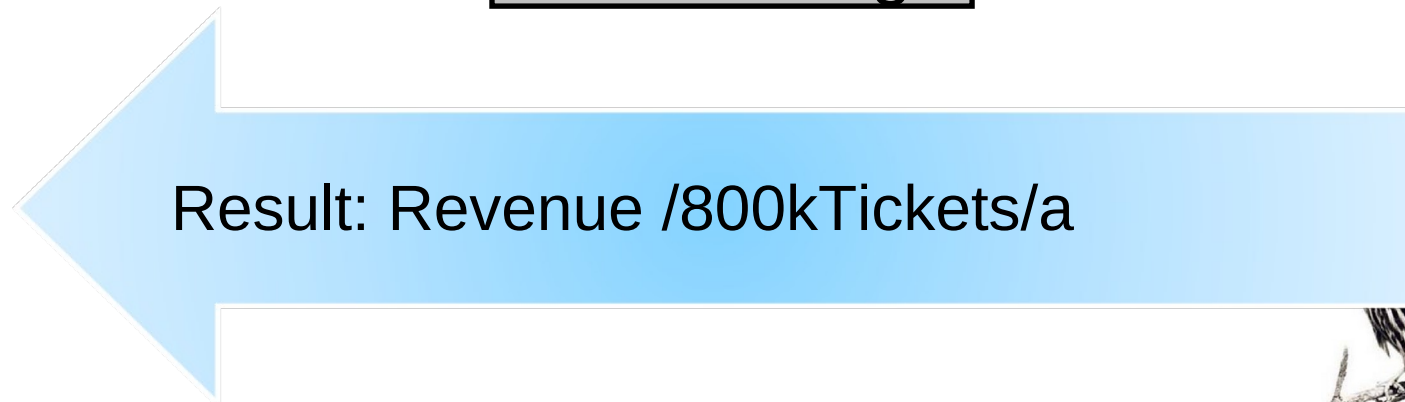




Measurement of closed Processes („Economies of Scale vs. Scope“)



Messung geschlossener Prozesse („Economies of Scale & Scope“)



Defizite klassischer Methoden

- Unnötige Inflexibilität aus der Methode heraus
 - Eine komplette Analyse eines komplexen Software-Projektes ist im Vorhinein nicht / unangemessen schwer möglich
 - Kunde will Preis und Dauer für das Gesamtprojekt wissen und erst dann den Auftrag erteilen
 - Anbieter kann erst nach gründlicher und vollständiger Analyse den Gesamtaufwand einschätzen
- Folgen:
 - Lange & aufwendige Analysen ohne Mehrwert
 - Umfangreiche Nacharbeiten
 - (zu) lange Projektlaufzeiten
 - Lösung passt nicht (mehr)
 - Anforderungs-Änderungen werden als Problem wahrgenommen
 - Die Doku (RDD) dient nur der späteren Schuldzuweisung



Scrum - Definition

- Dynamisches Vorgehensmodell zur Produkt-Entwicklung, insbesondere **Software**.
- Mittelpunkt: **kontinuierliche Weiterentwicklung** von Ressourcen, Prozessen & Werkzeugen
- Grundsätzliches Vorgehen:
 - Planung in **angemessener** Granularität
 - Arbeit in **selbst** organisierenden Teams
 - Nach Grundsätzen der **TOC**



Scrum – Vorgehen (1)

vor dem Sprint

- Anforderungen / Requirements werden in einer „**Anforderungsliste**“ / **Product-Backlog** (PBL) / (analog RDD) gesammelt, priorisiert, (je priorisierter, umso detaillierter) gepflegt & erweitert.
- Monatlich werden beim „Sprint Planning Meeting“ vom Team und dem Product Owner **definierte Arbeitspakete (Increments)** entnommen und in „**Sprints**“ (vergl. „frozen Zones“ & Staffelläufer-Prinzip) incl. Test und Doku umgesetzt.
- Der Rest des Product-Backlogs bleibt in (kontinuierlicher) Weiterentwicklung.



Scrum – Vorgehen (2)

beim Sprint

- Das Arbeitspaket (Increment) wird in kleinere Arbeitspakete (**Tasks**) mit definierten Bearbeitern herunter gebrochen und mit täglich aktualisiertem Restaufwand im „**Sprint-Backlog**“ verfolgt.
- Während des Sprints arbeitet das Team **ungestört** daran, die Tasks in einen produktiv einsetzbaren Anwendungsteil umzusetzen.
- Agenda des täglichen Abgleichs des Teams während eines 15min-**Daily-Scrum-Meetings**:
 - Was wurde von jedem erreicht? Status-Info für alle
 - Probleme? Issues; (->Task Manager)
 - Was steht bei jedem an? Abstimm-Info für alle
 - **Restdauer** akt. Sprint-Backlog
- Fortschritts-Dokumentation
 - TOC-Grafik Pufferverbrauch zu Fortschritt oder
 - „**Burndown-Chart**“



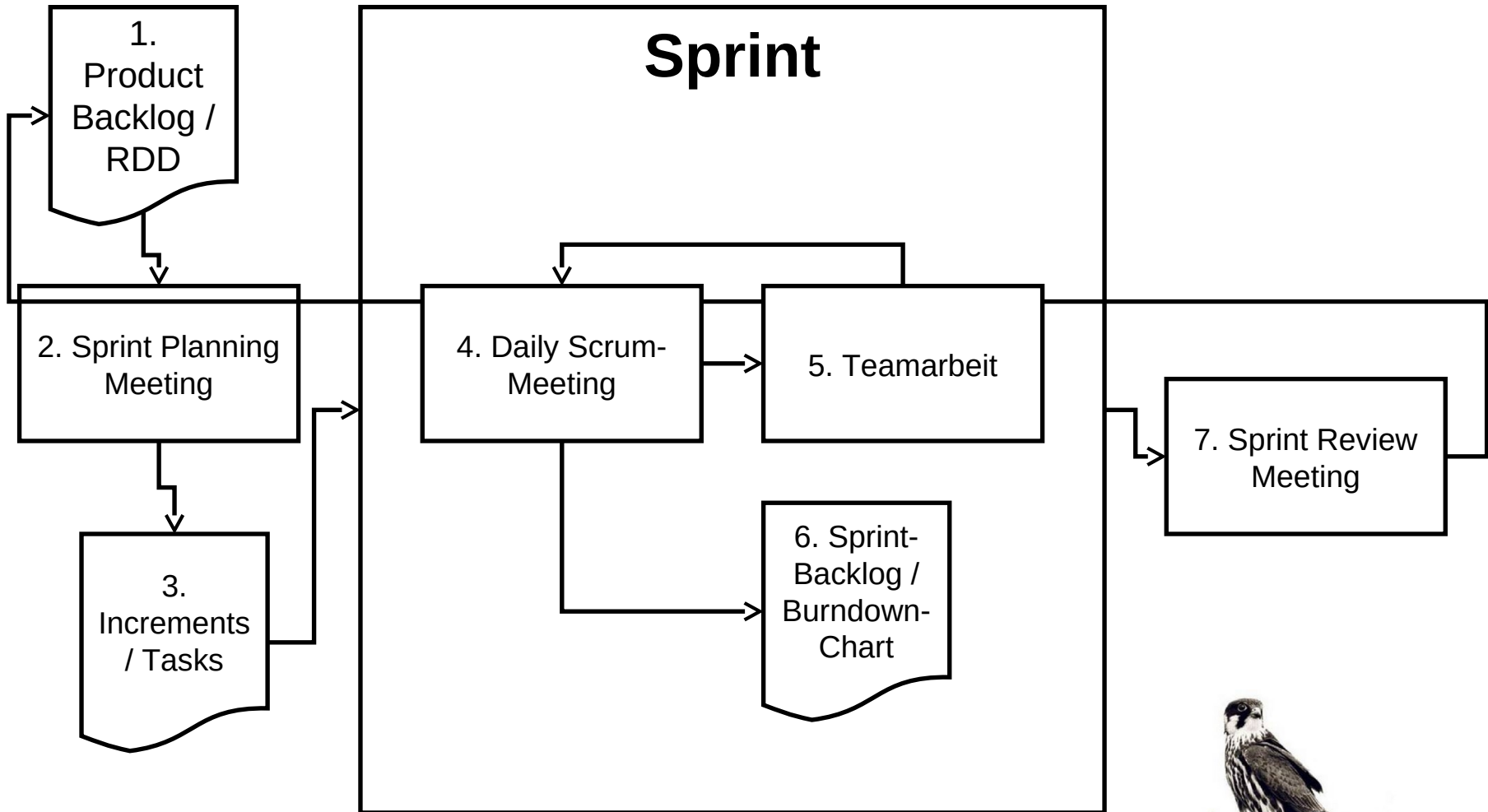
Scrum – Vorgehen (3)

nach dem Sprint („ist vor dem Sprint“)

- Das Team präsentiert dem Product Owner und den Stakeholdern im „**Sprint Review-Meeting**“ live am System die implementierte Funktionalität.
- Feedback & neue Anforderungen fließen dann über das Product Backlog in das nächste **Sprint Planning-Meeting** ein.



Scrum - Überblick



Scrum - Grundregeln

- Individuen gelten mehr als Prozesse und Tools
- Funktionierende Anwendungen gelten mehr als ausführliche Dokus
- Stetige Zusammenarbeit mit dem Kunden steht über Verträgen
- Offenheit für Änderungen steht über dem Befolgen eines festen Plans



Scrum - Rollen

- Product-Owner:
 - Fachliche Anforderungen aus User-Stories mit Prioritäten für „Sprints“
- Scrum-Master:
 - Prozess managen, Regel-Einhaltung überwachen, Hindernisse beseitigen
- Team & Stakeholder:
 - Team: schätzt Aufwand & entwickelt Produkt
 - Stakeholder: Beobachter und Ratgeber



Anhang



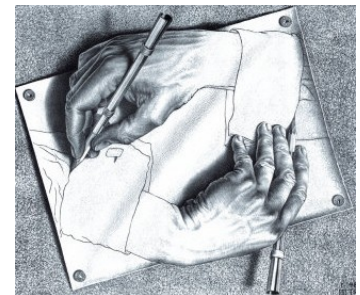
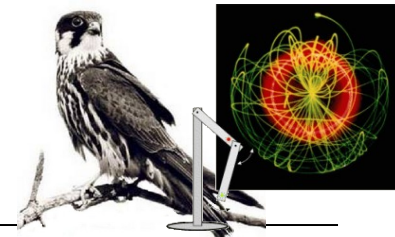
Abgrenzung Effizienz/Effektivität

Effizienz	Effektivität
die Dinge richtig tun	die richtigen Dinge tun
Verhältnis zwischen Ergebnis und eingesetzten Mitteln	Verhältnis vom Erreichten zum definierten Ziel
Wirtschaftlichkeit, Wirkungsgrad	Zielerreichungsgrad
Ziel mit minimalem Mitteleinsatz erreichen	gut erfüllte Aufgabe
mit minimalem Spritverbrauch von A nach B gelangen	Feuer löschen mit Champagner
Leistungsfähigkeit	Wirksamkeit
Taktik	Strategie
"How"	"What"
Management	Führung
Inhalt	Struktur



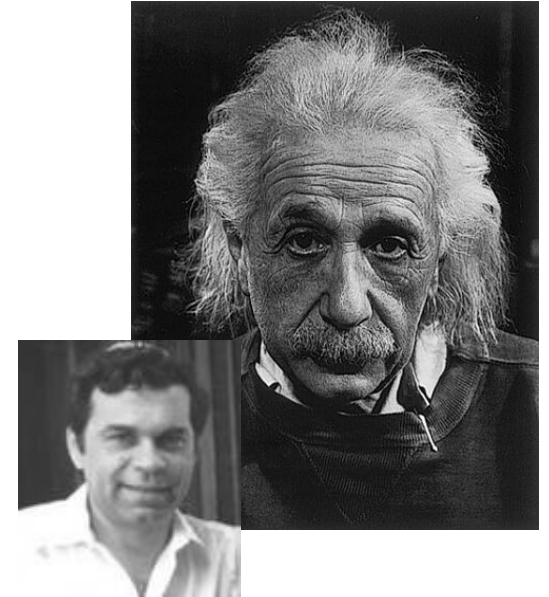
Abgrenzung Komplexität / Kompliziertheit

- Kompliziert = Gegenteil von einfach, im Gegensatz zu Komplexität keine Wechselwirkung zwischen den Elementen
- Komplex: probabilistisches / nicht determinierbares Verhalten, auch bei genauer Kenntnis der Einzelkomponenten und deren Wechselwirkungen

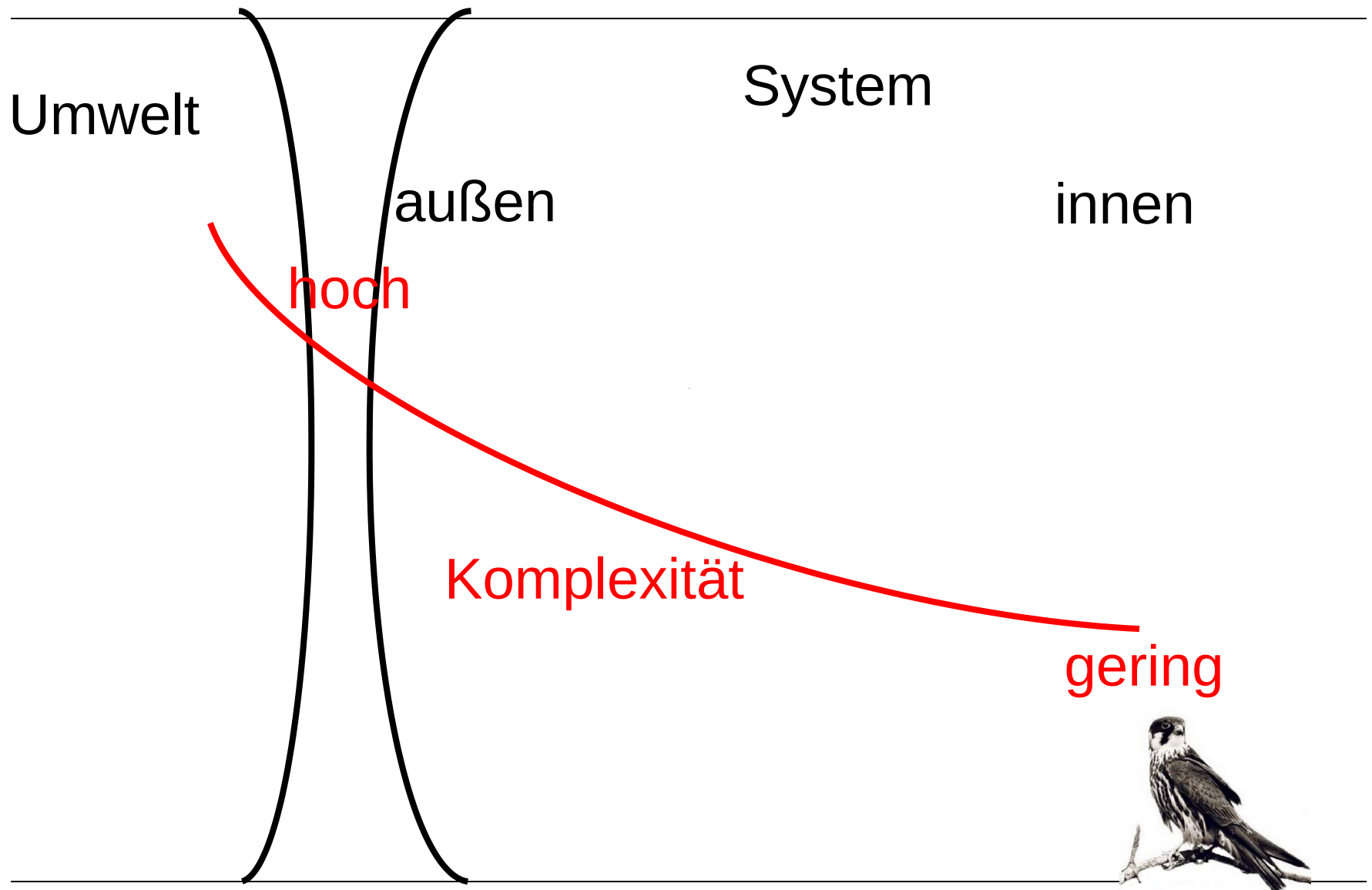


Umgang mit Komplexität

- Man sollte alles so einfach wie möglich machen, - aber nicht noch einfacher!
- Alle Systeme sind einfach (inherent Simplicity)
- Die Komplexität eines Systems sollte von aussen nach innen abnehmen; die Ordnung; Struktur zunehmen



Komplexitäts-Management



Die 3 Grundgesetze der TOC

- Alle Systeme sind einfach
- Alle Menschen sind Gut
- Es gibt keine Konflikte
 - Einem Konflikt liegt mindestens eine falsche Annahme zugrunde



Konzept- oder Personalqualität?

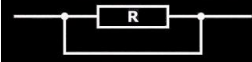
Konzeptqualität	Hoch	Hoher Förderungscharakter durch das System, gute Steuerbarkeit des Personals, hohe Wirtschaftlichkeit	Viele teure Systemveränderer, problematisch Konstanz und Kurs zu halten, angebracht in Early Stage zur Systementwicklung
	Gering	In jedem Falle Mist	Teure, überwiegend durch Geld halbwegs motivierbare Leute ohne Firmenbindung mit hoher Fluktuation und problematischer Steuerung
		Gering	Hoch

Personalqualität



Die Ebenen des Widerstands

Widerstand ist zwecklos



Widerstandsebene	Vorgehen	Hilfsmittel
1 Haben wir ein Problem?	Einsicht in die Zusammenhänge der Ues liefern	CRT; wenn, dann, Vesters PC
2 Und das soll das Problem lösen?	falsche Annahmen identifizieren, Widerstand in Mitwirkung verwandeln	Dilemma-Wolke(n) CCC
3 Das bringt doch nichts!	mit Vorwärts-Logik (wenn-dann) die Eee identifizieren	FRT; aus der Injektion
4 Ja, aber!	als Scrutinizing nutzen, negative Nebeneffekte konstruktiv aufnehmen, Stolpersteine in Steigbügel umwandeln	PRT
5 Schaffen wir nie!	Stolpersteine in Meilensteine umwandeln ("Wie" =How)	TT
6 keine Handlung trotz Zustimmung	Task-Management	Status-management
7 ?	?	?



Spielregeln im Meeting

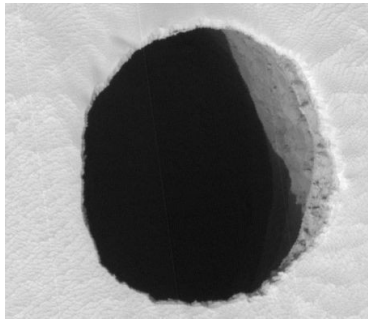


- Fehler sind erlaubt



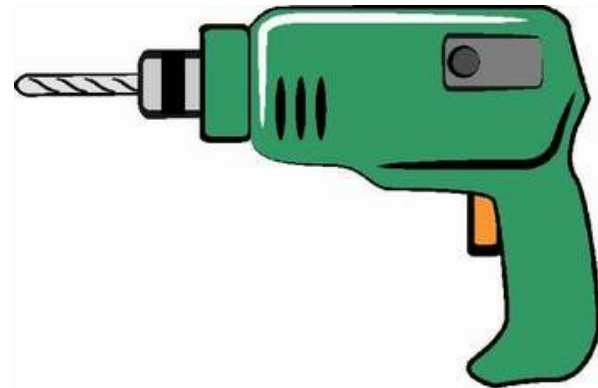
Der Unterschied

- Was es tut:



- Individuell
- Lastenheft
- Kundenwunsch

- Was es ist:



- Allgemein
- Pflichtenheft
- Produkt



sMart

1. No Improvement without Measurement

U can not improve what U can not measure

2. Measurement = Improvement

what U measure is already improved



Reifegrade

- 0 Unaware
- 1 Reactive problematic
- 2 Repetitive repeated
- 3 Aware defined
- 4 controlled measured
- 5 Proactive initiative

