



Information Management LE06 – Management der Anwendungen

Prof. Dr. Matthias Söllner

Universitätsprofessor für Wirtschaftsinformatik und Systementwicklung
Direktor am Wissenschaftlichen Zentrum für IT-Gestaltung (ITeG)

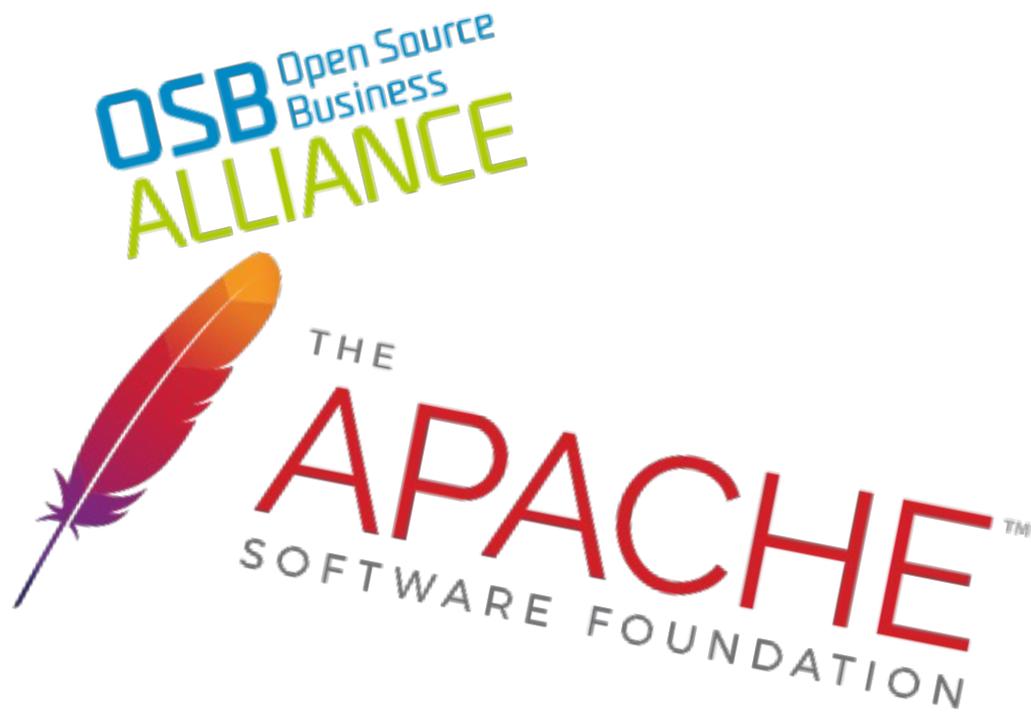
soellner@uni-kassel.de

www.uni-kassel.de/go/wise

Vorlesungsplan

Datum	Lerneinheit	
15.03.2024	Einführung & Grundlagen	Was will Mark Zuckerberg mit WhatsApp?
15.03.2024	Informationswirtschaft	Sind Informationen das Öl des 21sten Jahrhunderts?
18.03.2024	Informationsangebot	Ok Google...Ich bin krank. Was mache ich jetzt?
18.03.2024	Management der Daten	Wann schlägt Mensch Maschine?
19.04.2024	Management der Prozesse	Was geht in meinem Unternehmen eigentlich so vor?
19.04.2024	Management von Anwendungen	Warum für Software bezahlen, wenn es Open Source gibt?
22.04.2024	Innovative IKT	Warum gibt mein Chef mir nicht endlich ein Macbook Air?
22.04.2024	Wartung und Betrieb der IKT	Kann ein ehemaliges Staatsunternehmen überhaupt Innovativ sein?
03.05.2024	Speicherung und Kommunikation	Wird der FC Luzern doch der nächste Meister?
03.05.2024	Sicherheit und Organisation	Woher weiß ich, dass hier jeder nur das sieht, was er sehen soll?
06.05.2024	Führungsaufgaben	Wie kann ich alle IT-Themen unter einen Hut kriegen?
06.05.2024	Klausurvorbereitung	Was möchte ich nochmal erklärt haben?

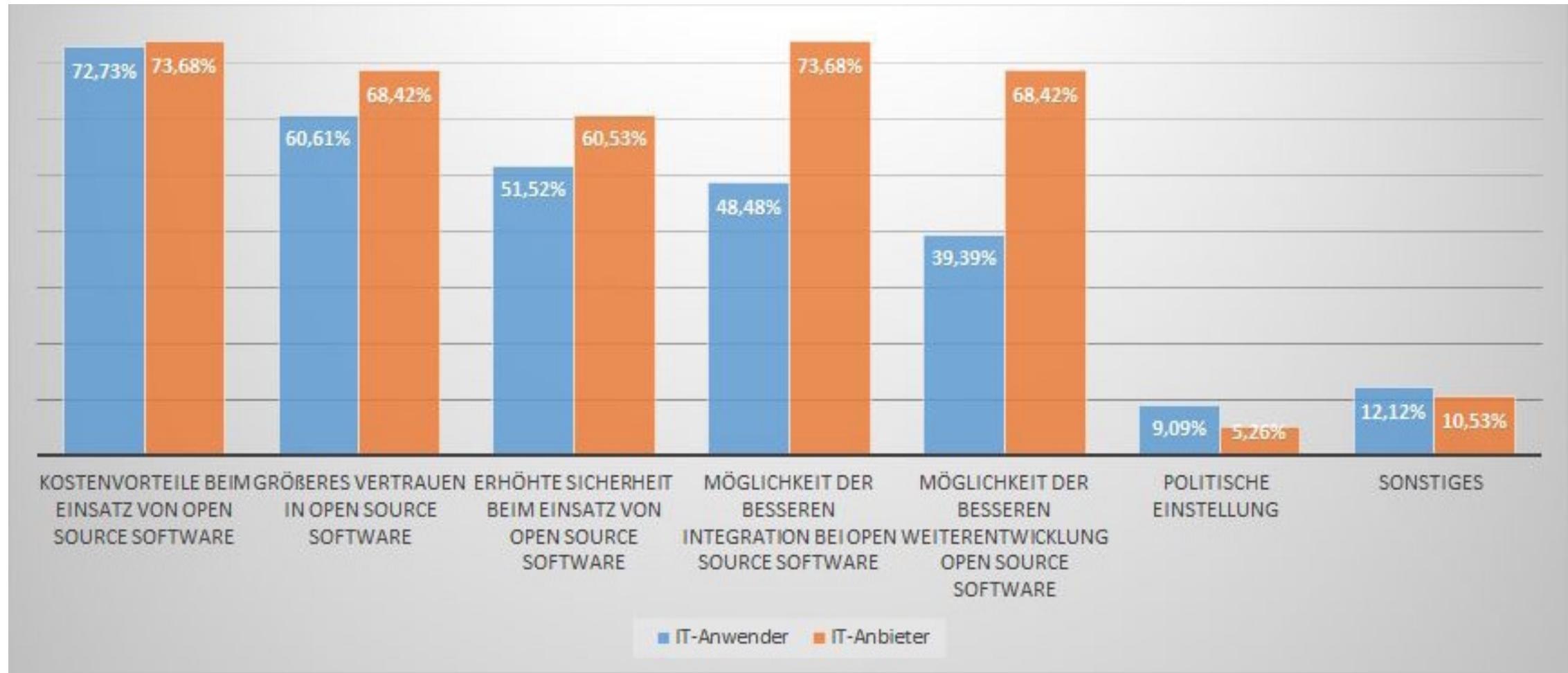
Sollte jeder sich für Open Source entscheiden?



Linux



Was begründet die OS Strategie eines Unternehmens?



Quelle: <http://osb-alliance.de/news/pressemitteilungen/osba-studie-open-source-ist-wirtschaftliches-schwergewicht>

Lernziele LE06 – Management der Anwendungen

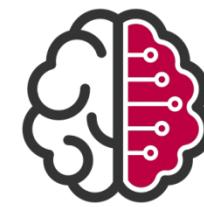


- 1) Sie kennen das Konzept des **Anwendungslbenszyklus** und die Aufgaben, die im Rahmen des Lebenszyklus anfallen.
- 2) Sie können Methoden zur **Softwareauswahl**, zur **Beurteilung von Lizenzmodellen** und **Softwareentwicklung** erläutern, beurteilen und anwenden.

Agenda LE04 – Management der Daten

- 
- 1 Management des Anwendungslebenszyklus
 - 2 Einführung von Software
 - 3 Gesamtheit der Anwendungen

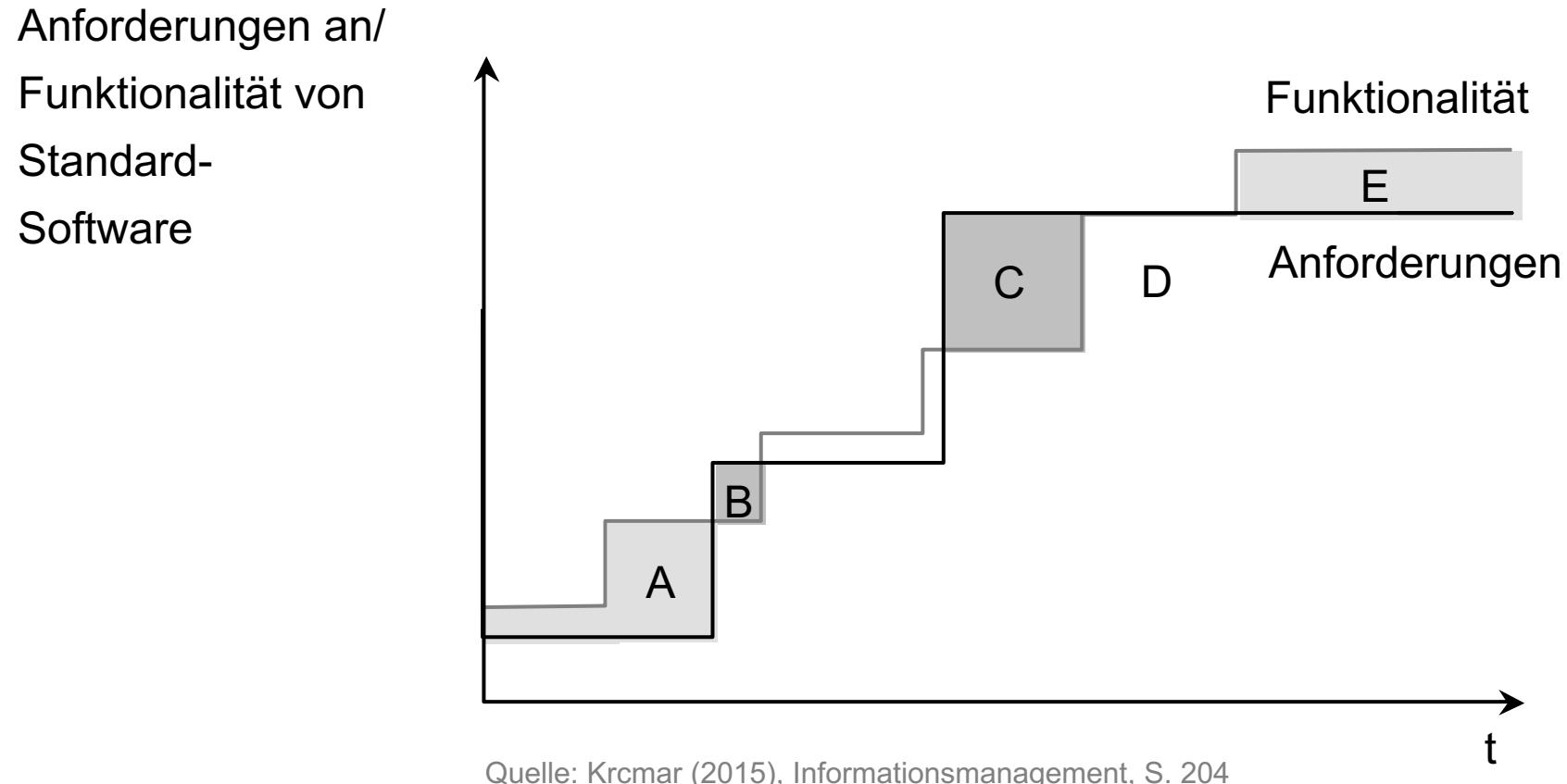
6.1. Management des Anwendungslebenszyklus



W I S E

6.1. Management des Anwendungslebenszyklus

Funktionalitätsbedarf und -angebot bei Standardsoftware



6.1. Management des Anwendungslebenszyklus

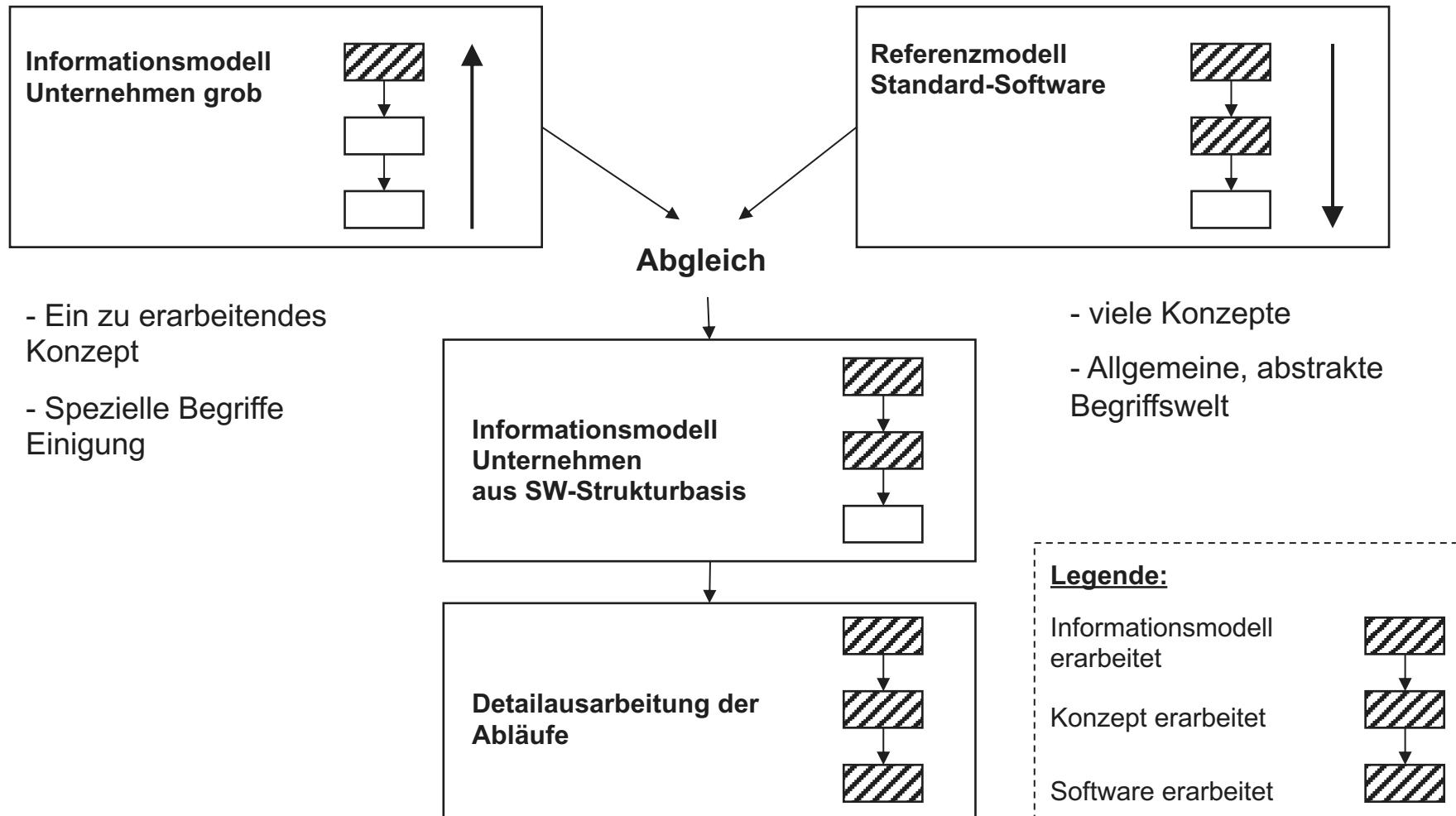
Anpassung von Standardsoftware an betriebliche Erfordernisse

Situation	Anpassungsweg	Probleme
Zusammenfallen	-	Entweder-oder-Entscheidung
Ausnutzen	Viele Funktionen Viele Tabellen	Auswahlproblem Komplexität
Dauerhafte Anpassung	Anpassung innen (Modifikation) Anpassung außen (Ergänzung durch Satellitensysteme)	Übernahme neuer Versionen Schnittstellen Wartungsaufwand
Fallspezifische Anpassung	Workflow Management	Beherrschbarkeit Erlernbarkeit

Quelle: Krcmar (2015), Informationsmanagement, S. 205

6.1. Management des Anwendungslebenszyklus

Modellorientierte Einführung von Standardsoftware



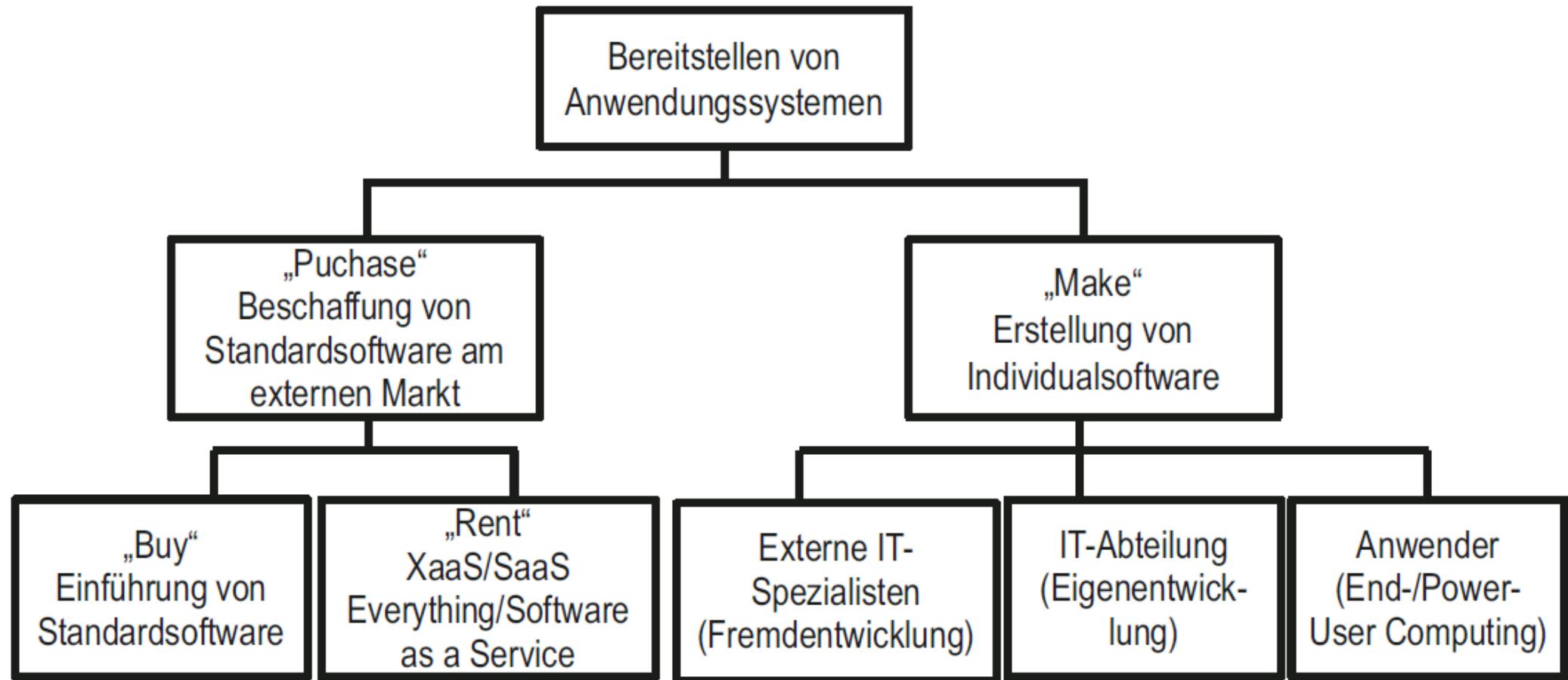
6.1. Management des Anwendungslebenszyklus

Anforderungen an Software

- Funktionalität
- Parametrisierbarkeit
- Kompatibilität zur installierten Basis
- hohe Datenverfügbarkeit
- Benutzerfreundlichkeit (Aufgabenangemessenheit, Selbstbeschreibungsfähigkeit, Steuerbarkeit, Erwartungskonformität, Fehlerrobustheit)
- Effizienz
- Sicherheit
- Anschaffung (Lieferzeit, Kaufpreis, Wartung, ...)
- Anbieter (Service-Zentrum, Qualifikation, Referenzkunden, ...)

6.1. Management des Anwendungslebenszyklus

Alternativen der Software-Bereitstellung



Quelle: In Anlehnung an Schwarzer (2014), Krcmar (2015), Informationsmanagement, S. 212

6.1. Management des Anwendungslebenszyklus

Kategorisierung der Software-Entwicklung



Quelle: In Anlehnung an Brügge und Dutoit (2006), Krcmar (2015), Informationsmanagement, S. 223

6.1. Management des Anwendungslebenszyklus

Vor – und Nachteile von Standardsoftware

Vorteile Standardsoftware	Nachteile Standardsoftware
Kosteneinsparungen	Unvollständige Abdeckung der Anforderungen
Schnellere Verfügbarkeit und Nutzung	Probleme bei Integration in bestehende Strukturen
Hohe Programmspezialität durch spezialisierten Anbieter	Orientierung an allgemeiner Verwendbarkeit → evtl. schlechtes Betriebsverhalten in unternehmensspezifischen Situationen
Wartung und Weiterentwicklung durch Anbieter	
Unabhängig von Grösse und Verfügbarkeit der internen IT-Ressourcen	

6.1. Management des Anwendungslebenszyklus

Vorteile von Open-Source-Software (OSS)

- Höhere Verbreitung, höhere Qualität und Stabilität
- An eigene Bedürfnisse besser anpassbar
- Geringere Herstellerabhängigkeit, da die SW von jedem weiterentwickelt werden kann
- Höhere Sicherheit durch Einsehbarkeit der Quellcodes
- Keinerlei Kosten zur Lizenzierung, Transaktionskosten verringern sich

6.1. Management des Anwendungslebenszyklus

Software-Lizenzmodelle

Modelltyp und primäre Bezugsgröße	Ausgewählte Beispiele für Lizenzmodelle in der Praxis
Primär nutzerbezogene Modelle: Anzahl der Nutzer	Definierte Nutzer vs. indirekter Zugriff Lizenzkosten pro Bürger Lizenzkosten während Evaluation
Primär wertbezogene Modelle: z. B. Personalbestand oder Herstellungskosten der verkauften Produkte	Lizenzen für Personaladministrations-SW Lizenzen für Planungs- und Dispositionssysteme
Primär zeitbezogene Modelle: Dauer der Nutzung	Subskription (Abonnement)
Primär infrastrukturbbezogene Modelle: Ausmaß der Nutzung der genutzten Infrastruktur	Pro-Device-Lizenz Abrechnung nach Prozessor- oder Speichernutzung

Quelle: Krcmar (2015), Informationsmanagement, S. 224

6.1. Management des Anwendungslebenszyklus

Grundlagen des SAP-Lizenzmodells

- Subskriptionsbasierte Lizenz:
 - Diese ist ausschließlich für **Cloud-basierte SAP-Lösungen** verfügbar
 - Diese ermöglicht einem einen **zeitlich beschränkten Zugriff** auf SAP-Lösungen
 - Am Anfang hat man eine bestimmte **Mindestlaufzeit**, anschließend muss die Lizenz in **regelmäßigen Abständen erneuert** werden
- Kontinuierliche Lizenz
 - Grossteil der SAP Lizenzen sind kontinuierliche Lizenzen
 - Diese ermöglicht einem einen **zeitlich unbeschränkten Zugriff** auf SAP-Lösungen
 - Sie besteht aus **Paketlizenzen, Nutzerlizenzen** und einem **Wartungsvertrag**

6.1. Management des Anwendungslebenszyklus

Bestandteile der kontinuierlichen Lizenz

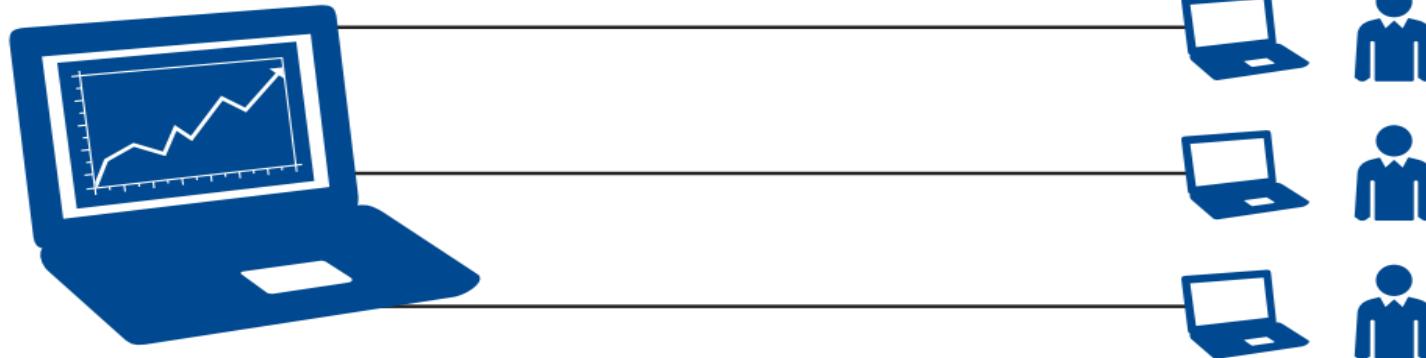
Paketlizenz

- Verwendung bestimmter Funktionalitäten
- Basiert auf Geschäftskennzahlen



Nutzerlizenzen

- Nutzer nutzt lizenzierte Funktionalitäten
- Richtet sich nach der Anzahl der Nutzer



Beide Lizenzen können **nicht unabhängig** voneinander verwendet werden

Wartung

- Entspricht einem bestimmten %-Betrag der gesamten Lizenzkosten

6.1. Management des Anwendungslebenszyklus

Kriterien für die Bewertung von Lizenzmodellen

- Geringe Initialkosten → Subskriptions-Lizenzmodell
- Testphase → userbezogene Lizenzierung
- Gemeinsamer Zugriff mehrerer User → Pro-Device-Lizenzen
- Flexible Infrastruktur → infrastrukturbbezogenes Software-Lizenzmodell
- Regionale oder branchenspezifische Besonderheiten
→ wertbasiertes Lizenzmodell
- Application Service Provider oder Cloud Provider → Service-Provider-Lizenzmodelle

6.1. Management des Anwendungslebenszyklus

Frage 1:

Sie sind CIO einer großen Marketingfirma. Bei bestimmten Marketingkampanien benötigen ihre Mitarbeiter eine spezielle Standardsoftware zur Videobearbeitung. Welches Lizenzmodell bevorzugen Sie?

- A Nutzerbezogene Modelle
- B Wertbezogene Modelle
- C Zeitbezogene Modelle
- D Infrastrukturbbezogene Modelle

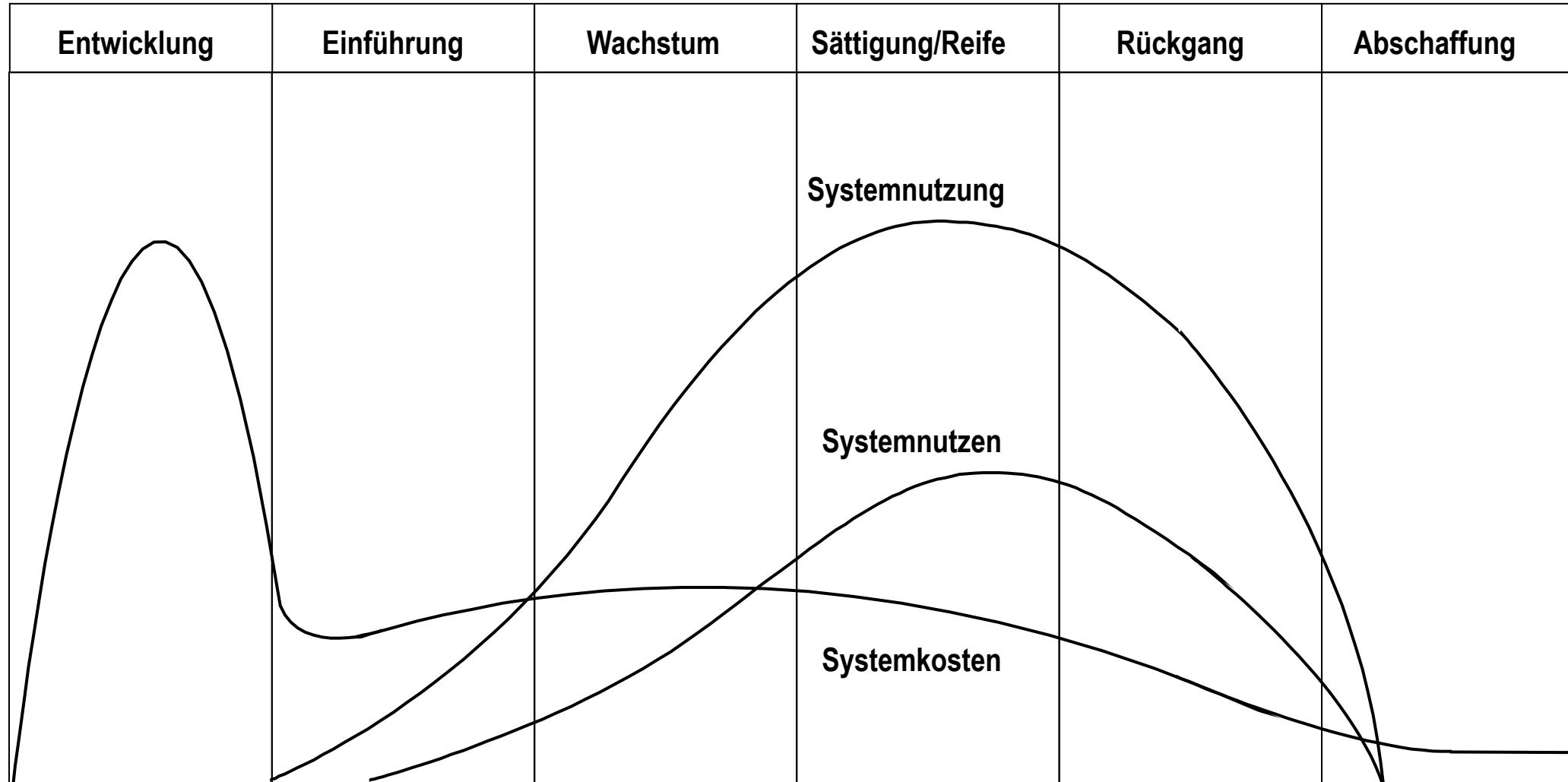
6.1. Management des Anwendungslebenszyklus

Lebenszyklus?



6.1. Management des Anwendungslebenszyklus

Lebenszyklus-Modell

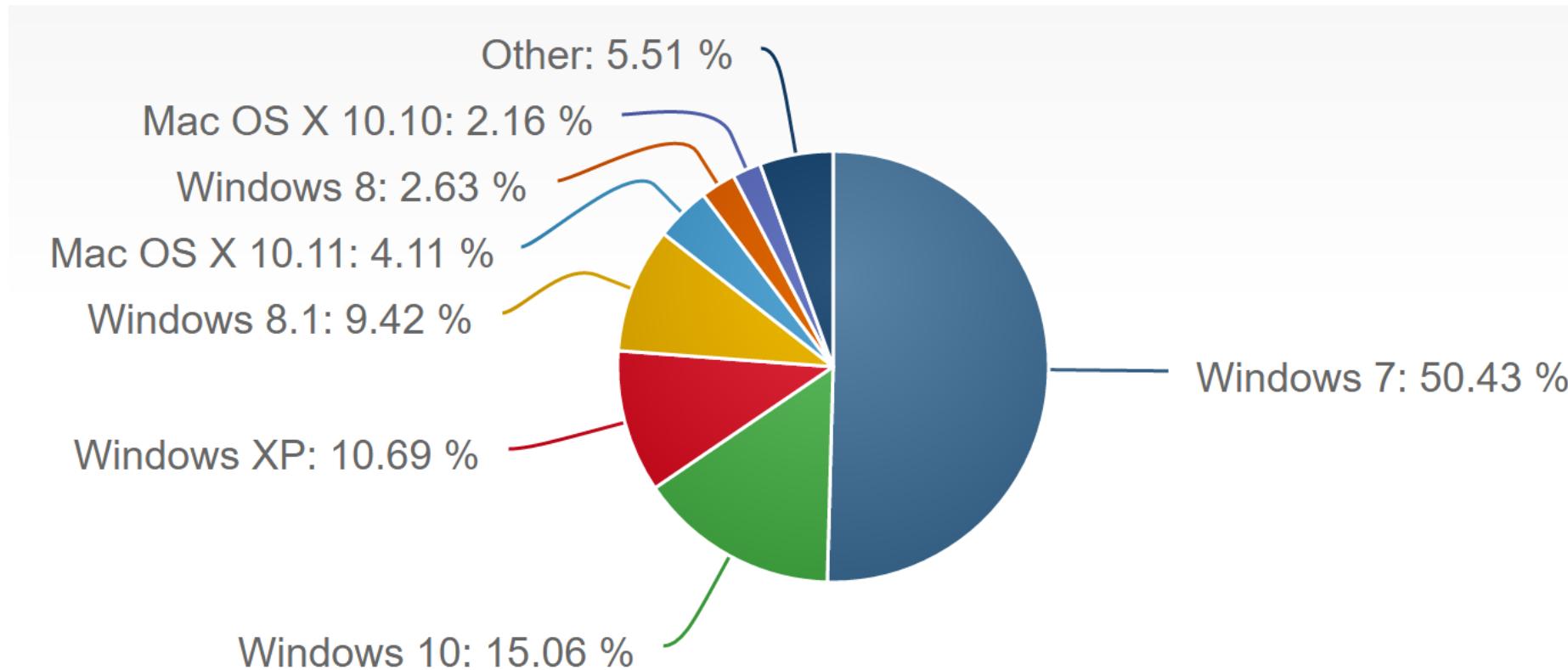


Quelle: In Anlehnung an Heinrich (2002), Krcmar (2015), Informationsmanagement, S. 229

6.1. Management des Anwendungslebenszyklus

Das Ende von Windows XP – Ein Paradies für Hacker?

- Weltweite Verbreitungsrate von Desktop-Betriebssystemen (April 2017):



Quelle: <http://marketshare.hitslink.com/operating-system-market-share.aspx>, zugegriffen am 15.05.2017

6.1. Management des Anwendungslebenszyklus

Weiterhin hohe Verbreitungsrate von Windows XP

- Geldautomaten
 - 95% aller Geldautomaten laufen mit Windows XP
 - Berlin, Oktober 2015: Mann knackt Geldautomat mit USB Stick
- Windows XP ist sehr anfällig für Malware
 - Ransomware WannaCry verbreitete sich sehr stark im Mai 2017
 - Microsoft sah sich genötigt einen Patch für Windows XP anzubieten, obwohl der Support schon seit April 2014 endete

6.1. Management des Anwendungslebenszyklus

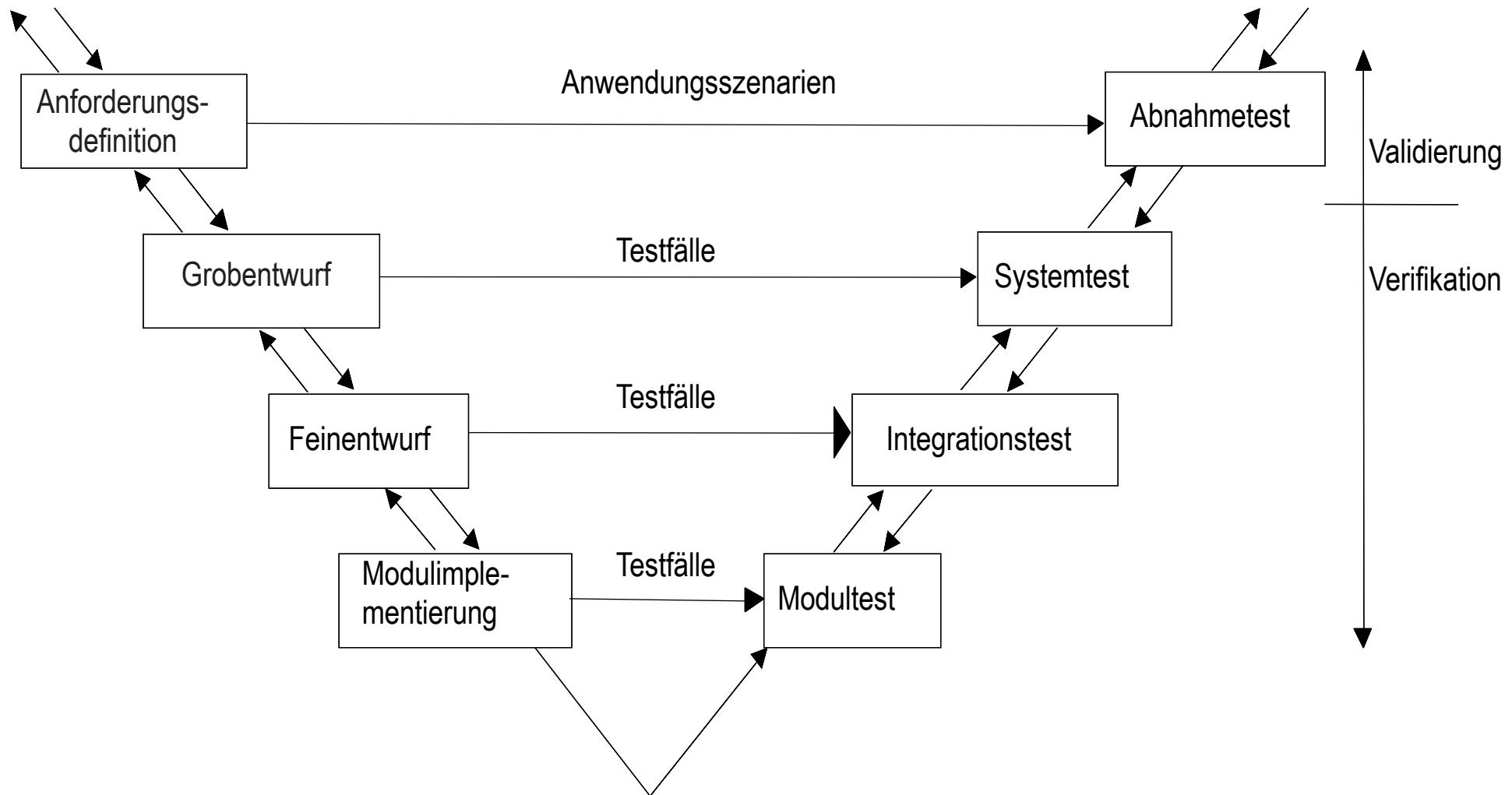
Überblick der Vorgehensmodelle zur Anwendungsentwicklung

		Ablaufgestaltung	
		Sequentiell	Iterativ
Formali-sierung	Stark formalisiert	Wasserfallmodell, V-Modell (200x), W-Modell, Inkrementell strukturierter Ansatz, (Neo-) Hermes ...	Spiralmodell, Rational Unified Process, Prototyping, OO Lifecycle-Modell, Feature Driven Development ...
	Wenig formal		Extreme Programming, Object Engineering Process, Partizipative Softwareentwicklung, Scrum ...

Quelle: Krcmar (2015), Informationsmanagement, S. 231

6.1. Management des Anwendungslebenszyklus

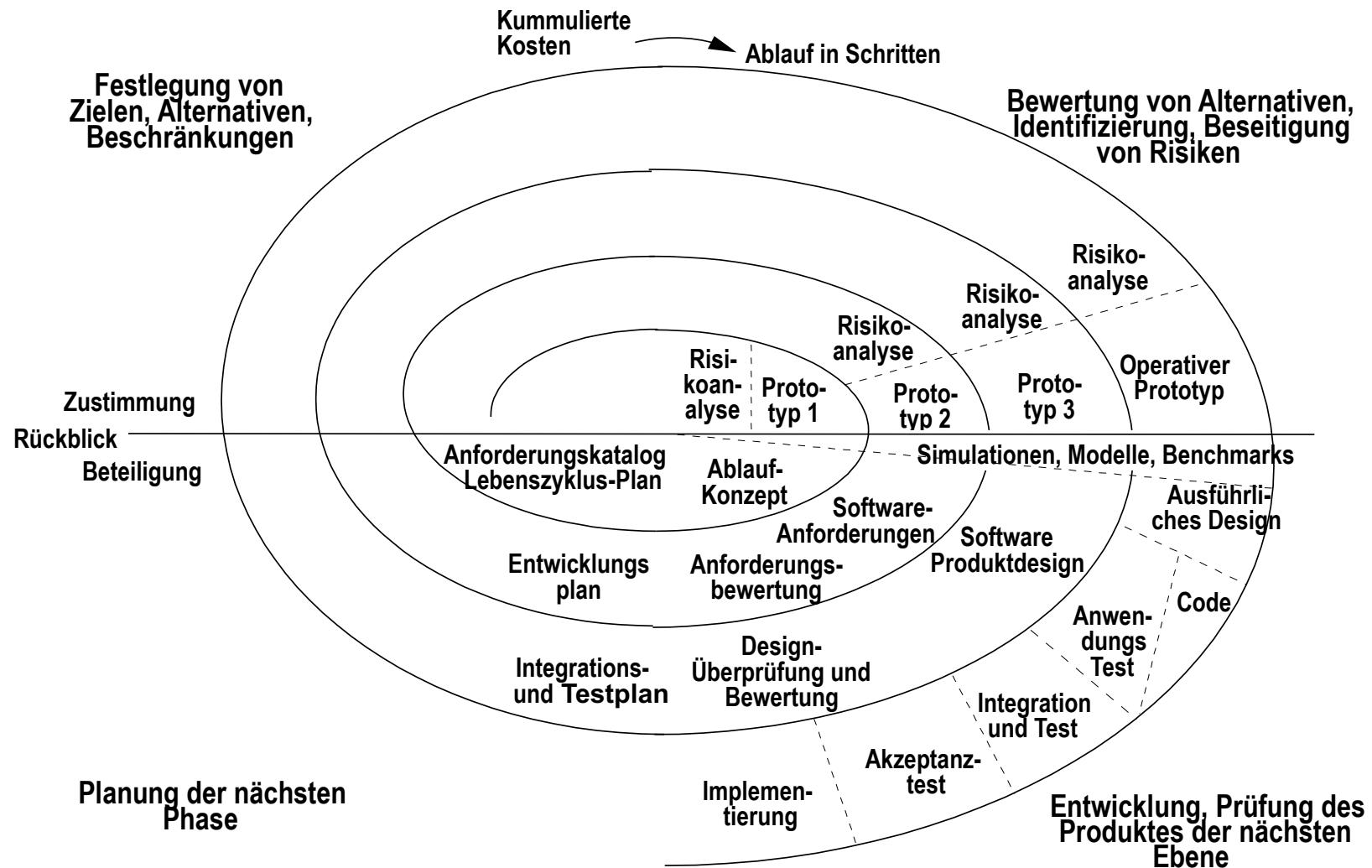
Das V-Modell



Quelle: Balzert (2002), Krcmar (2015), Informationsmanagement, S. 232

6.1. Management des Anwendungslebenszyklus

Spiralmodell der Softwareentwicklung



Quelle: Nach Boehm (1988), Krcmar (2015), Informationsmanagement, S. 234

6.1. Management des Anwendungslebenszyklus

Prinzipien agiler Softwareentwicklung

Individuen und Interaktionen

sind wichtiger
als

Prozesse und Tools

Funktionierende Software

ist wichtiger
als

Umfangreiche
Dokumentation

Kooperation mit
Projektbetroffenen

ist wichtiger
als

Vertragsverhandlungen

Reaktion auf
Änderungen

ist wichtiger
als

Verfolgung eines festgelegten
Plans

6.1. Management des Anwendungslebenszyklus

Agile Software-Entwicklung – Beispiel SCRUM

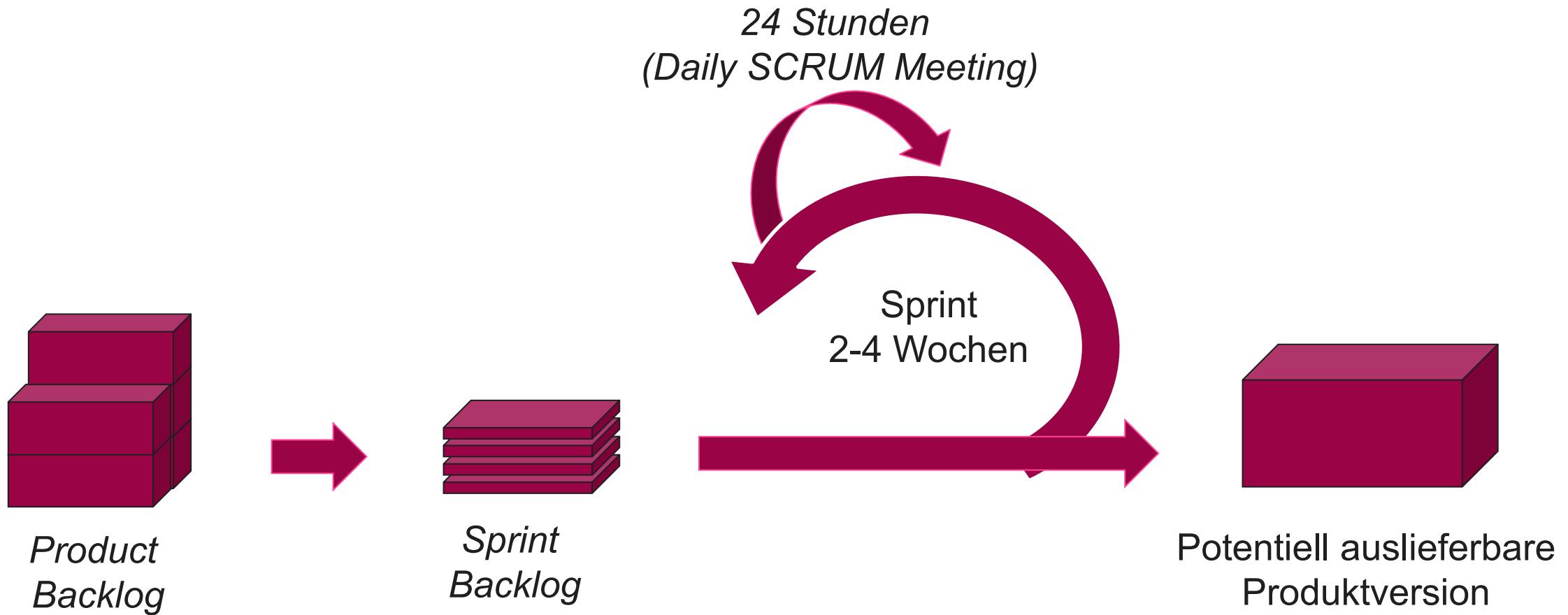
SCRUM

- Vorgehensmodell, aber auch Grundüberzeugung und Philosophie
- Komplexe Entwicklungsprozesse nicht im Voraus exakt planbar

- Selbstorganisierendes Team:
 - Scrum Master: sorgt für Einhaltung SCRUM-Prinzip, Planungsvorgaben kontrollieren
 - Product Owner: legt Entwicklungsziel fest, verwaltet Budget
 - Team: = Mitarbeiter; selbstorganisierte Aufgabenverteilung, Realisierungszeitpunkt festlegen
 - Product Backlog: dort werden alle Aufgabenpakete aufgeführt

6.1. Management des Anwendungslebenszyklus

Agile Software-Entwicklung – Beispiel SCRUM



6.1. Management des Anwendungslebenszyklus

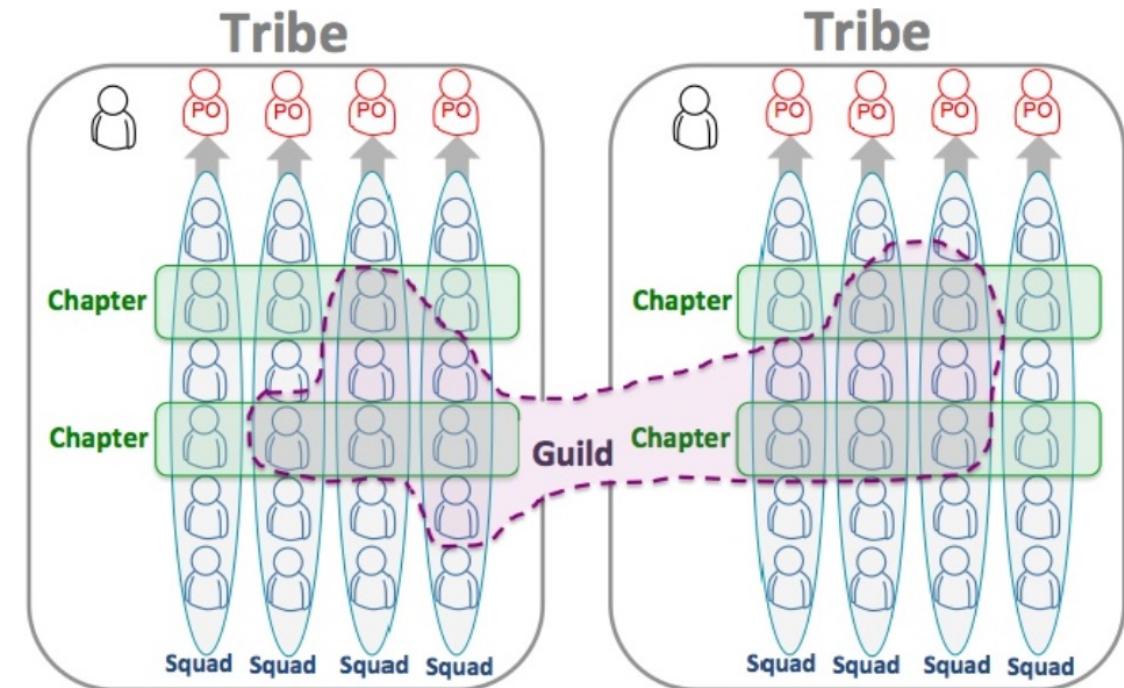
Agile Techniken – Beispiel SCRUM

- Daily Standup:
 - Kurzes, tägliches Status-Meeting
 - „Was habe ich gemacht, was plane ich zu tun?“
- Sprint Planning:
 - Festlegung der zu entwickelnden Anforderungen
 - Festlegung der Art und Weise der Umsetzung
- Sprint Retrospektive:
 - Kritische Hinterfragung der Arbeitsweise
 - Planung von Verbesserungsmaßnahmen

6.1. Management des Anwendungslebenszyklus

Fallbeispiel: Agile Entwicklung bei Spotify

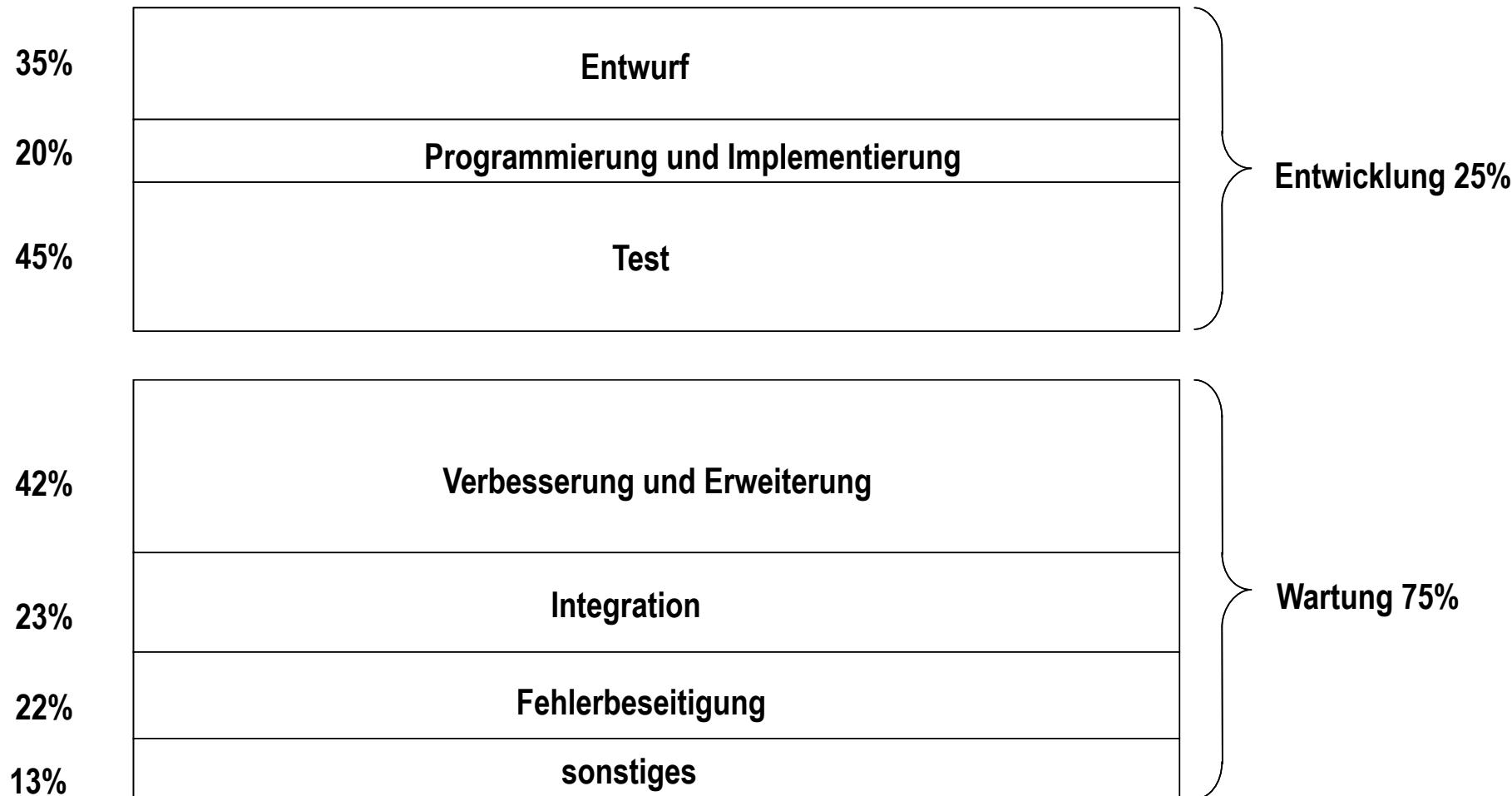
- Squad:
 - kleinste Gruppe
 - Ähnlich zu einem Scrum-Team
 - Selbstorganisierend und autonom
- Tribe: Squads, die in ähnlichen Bereichen arbeiten
- Chapter: Leute eines Kompetenzbereichs eines Tribes mit ähnlichen Fähigkeiten
- Guild: Gemeinschaft mit gleichen Interessen



Quelle: <https://techcrunch.com/2012/11/17/heres-how-spotify-scales-up-and-stays-agile-it-runs-squads-like-lean-startups/>, aufgerufen am 20.01.2017

6.1. Management des Anwendungslebenszyklus

Software-Kosten-Relationen bei Eigenentwicklung



Quelle: Scheer (1998), Krcmar (2015), Informationsmanagement, S. 247

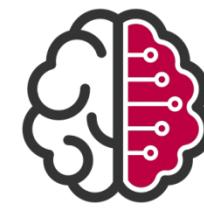
6.1. Management des Anwendungslebenszyklus

Kriterien zur Wahl des Kostenschätzungsverfahrens

Kein „ultimatives“ Verfahren, nur Tendenzen sind anzugeben; abhängig von gewissen Projektdaten:

- Zwang zur Durchführung des Projektes => Tragfähigkeitsmethode
- Projektstadium: frühe Phase => Use Case/Top-Down-Kostenschätzung
- Experten vorhanden => Expertenschätzung
- Projekt-Vergleichsbasis steht zur Verfügung
- Produktanforderungen bekannt => Analogieverfahren
- Einsatz wieder verwendbarer Softwarekomponenten / Einsatz standardisierter Middleware => COCOMO II
- Objektorientierte Entwicklung => Schätzung auf Basis von Object Points
- Falls alle diese Verfahren nicht möglich sind: Parkinson-Methode (Schätzung nach Verfügbarkeit der Ressourcen)

6.2 Einführung von Software



W I S E

6.2 Einführung von Software Projekt – Produkt – Prozess

- Ein **Projekt** ist ein Vorhaben, dass im Wesentlichen durch Einmaligkeit der Bedingungen gekennzeichnet ist, wie z.B.
 - Zielvorgabe,
 - Zeitliche, finanzielle und andere Begrenzungen,
 - Abgrenzung gegenüber anderen Vorhaben,
 - Projektspezifische Organisation.
- **Produkt**
 - als materieller oder immaterieller Gegenstand, der als Erzeugnis/ Ergebnis eines Tätigkeitsvorhabens bezeichnet werden kann und den Output von Projekten darstellt
- **(Geschäfts-)Prozess**
 - als sachlich-logische Abfolge von Unternehmensverrichtungen, die der Leistungserstellung dienen

6.2 Einführung von Software

Alternativen der SW Einführung

■ **Stichtagsumstellung**

- Totale Umstellung vom vorherigen Zustand (kein System oder altes System) auf die neue Anwendung zu einem Zeitpunkt

■ **Parallelisierung**

- Übergangsweise gleichzeitige Nutzung der alten (keine/alte Anwendung) und neuen Methoden bis zur Sicherstellung der Funktionsfähigkeit

■ **Teilweise Einführung**

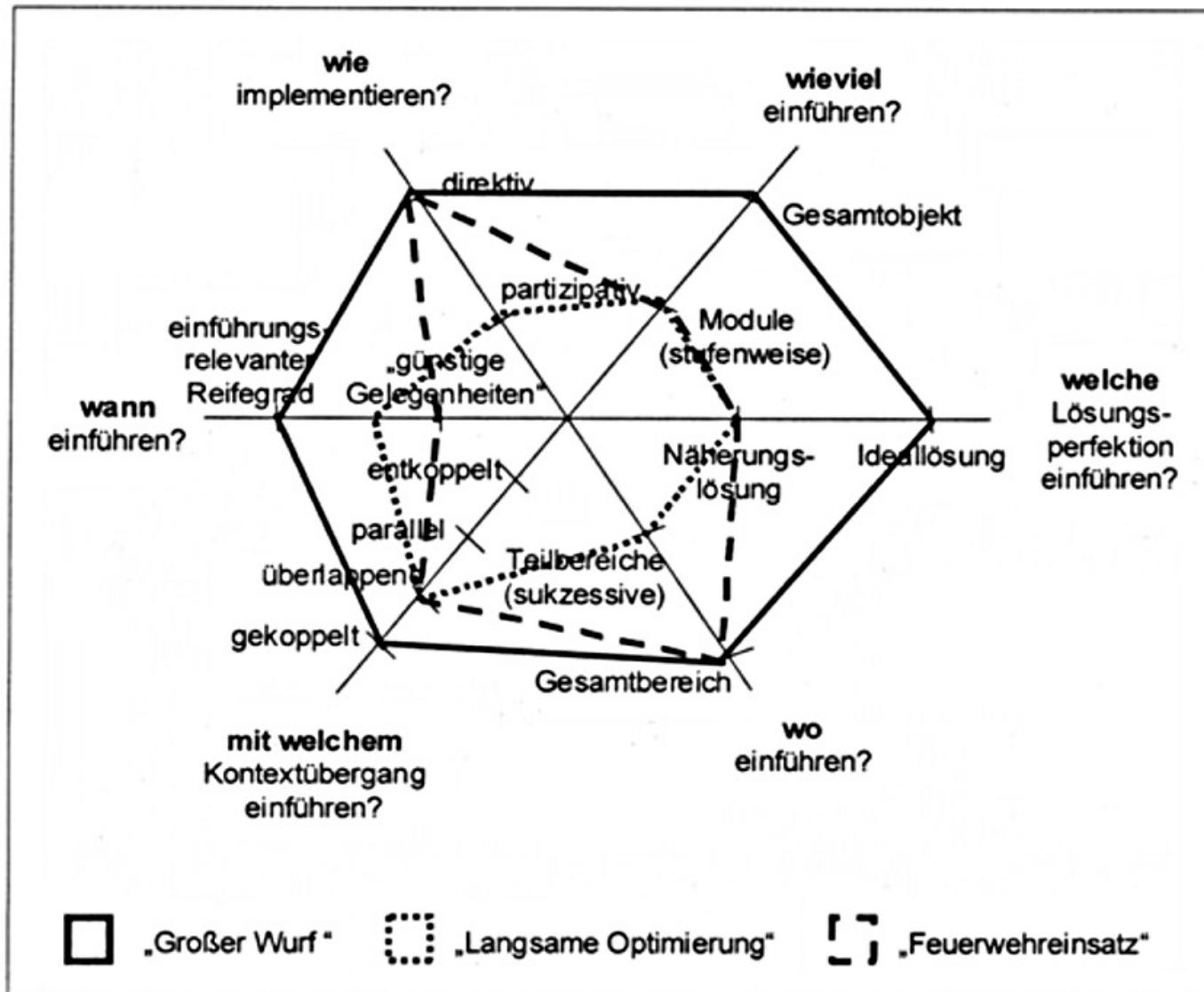
- Durchführung einer Stichtagsumstellung für einzelne Teile der neuen Anwendung durch Parallelisierung und allmähliche Ausweitung nach Sicherstellung der Funktionalität

■ **Versionsumstellung**

- Um zu einer wesentlich höheren Version einer Software zu gelangen, die einer gravierenden Änderung gleich kommt, kann der Umweg über niedrigere Versionen notwendig sein

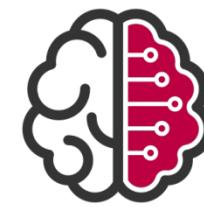
6.2 Einführung von Software

Spinnwebdiagramm ausgewählter Implementierungsstrategien



Quelle: Daniel (2001, S. 176), Krcmar (2010), Informationsmanagement, S. 274

6.3 Gesamtheit der Anwendungen



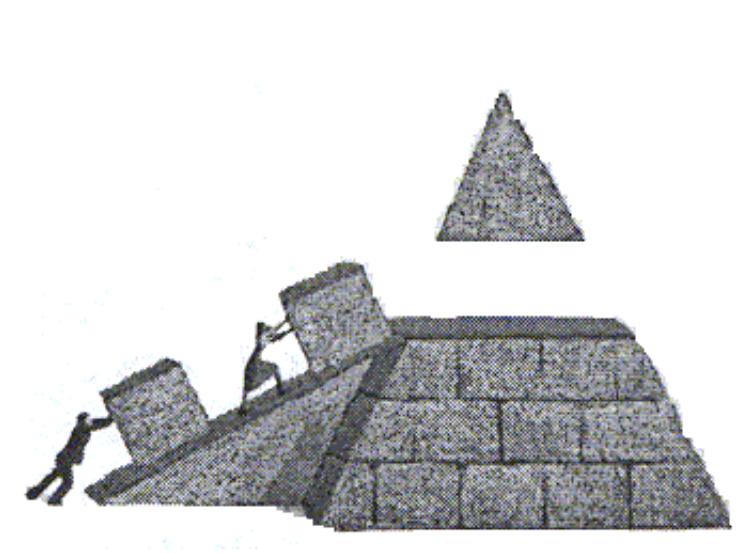
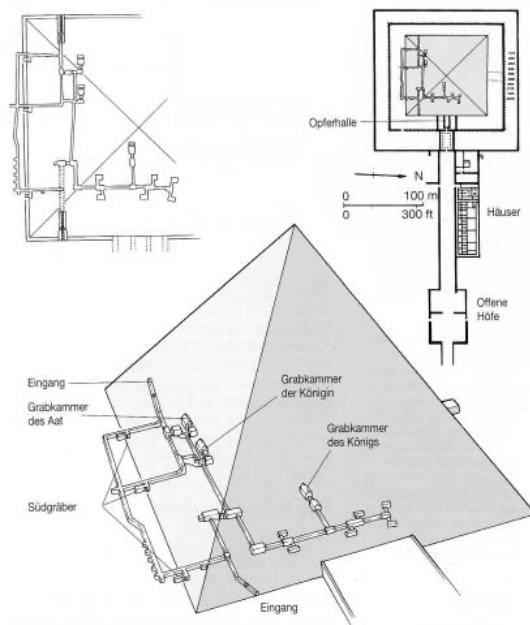
W I S E

6.3 Gesamtheit der Anwendungen

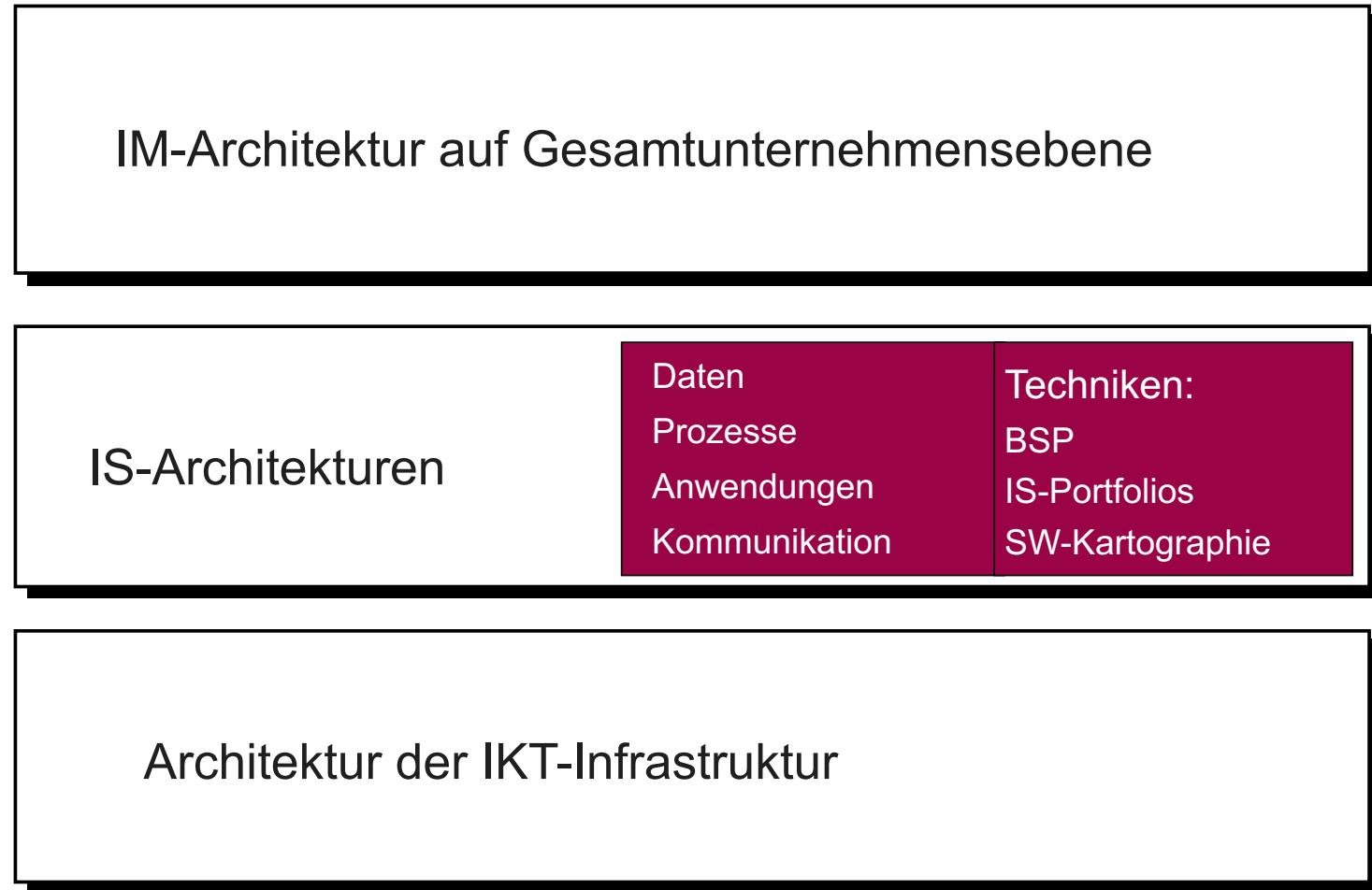
Definition: Architektur

„Generically, an architecture is the description of the set of components and the relationship between them.“

(Quelle: Armour, F.J.; Kaisler, S.H.; Liu, S.Y. (1999). A big picture look at an Enterprise Architectures. IEEE IT Professional 1 (1). S.37)



6.3 Gesamtheit der Anwendungen Einordnung von IS-Architekturen



6.3 Gesamtheit der Anwendungen

Definition IS-Architektur

“A **software architecture** describes the **layout** of the **software modules** and the **connections** and **relationships among them**. A **hardware architecture** can **describe how the hardware components are organized**.“

(Quelle: Armour, F.J.; Kaisler, S.H.; Liu, S.Y. (1999). A big picture look at Enterprise Architectures. IEEE IT Professional 1(1). S.37)

„Eine **IS-Architektur** ist die **strukturierende Abstraktion** existierender oder geplanter **Informationssysteme**.“

(Quelle: Dern (2003), S.18, Krcmar (2015), Informationsmanagement, S.297)

Ziel der Entwicklung einer IS-Architektur

- Gemeinsame Kommunikationsplattform aller an der Gestaltung von Informationssystemen Beteiligten zu schaffen
- Damit wird die Planbarkeit und die Steuerbarkeit der Gestaltung der betrieblichen Anwendungslandschaft erhöht

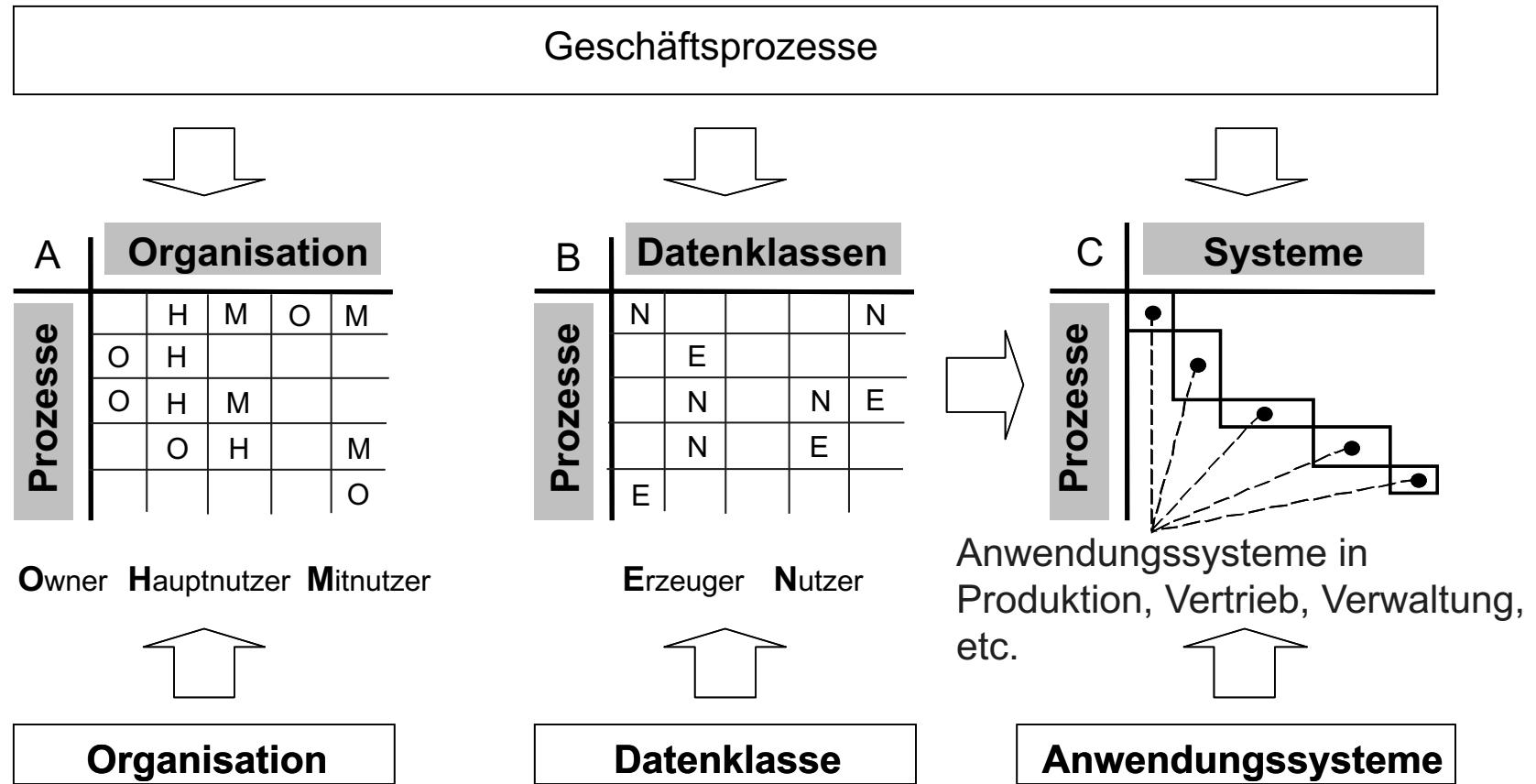
6.3 Gesamtheit der Anwendungen

Business Systems Planning (BSP)

- Von IBM ursprünglich in den 60er Jahren entwickelte Technik zum Ermitteln stabiler IS-Strukturen in Unternehmen
- Kennzeichen
 - Top-Down-Planung mit Bottom-Up-Implementierung
 - Behandlung von Daten als Unternehmensressource
 - Generelle Ausrichtung auf Geschäftsprozesse
 - Aufteilung in unternehmensweite und geschäftsbereichsweite Analyse
- Grundelemente
 - **Geschäftsressourcen** (Bsp. Kunden, Lieferanten, Bedarf, Kapital,...)
 - **Geschäftsprozesse** (oftmals in zwei- bis dreistufige Hierarchien aufgeteilt, Bsp. Personal einstellen, mit den Subprozessen Vorstellungstermin vereinbaren, Einstellungsverhandlungen führen, Personalakte anlegen,...)
 - **Datenklassen** (logisch zusammengehörende Informationen, die in einem Geschäftsprozess erzeugt und in einem oder mehreren Prozessen verwendet werden, Bsp. Kunden-Stammdaten, Kunden-Umsatzdaten,...)

6.3 Gesamtheit der Anwendungen

Die Matritzen einer BSP-Studie



6.3 Gesamtheit der Anwendungen

Risiko-Nutzen-Portfolio

- Übertrag des Portfolio- Ansatzes auf das IM als verständliches Verfahren zur Auswahl von IS-Anwendungen durch Cash/McFarlan/McKenney (1992)
- Gegenüberstellung von Risiko und Nutzen eines Projektes
- Risikokriterien
 - Misserfolg beim Erzielen aller oder auch nur einiger Nutzenkriterien
 - Höhere Implementierungskosten als erwartet
 - Längere Implementierungszeit als erwartet
 - Erheblich geringere technische Performanz als erwartet
 - Inkompatibilitäten des Systems mit bestimmter Hardware und Software

6.3 Gesamtheit der Anwendungen

Phasen der Erstellung eines IS Portfolios

1) Definition der Analysebasis

Untersuchung aller Anwendungen oder nur Teil-IS, auch für geplante Anwendungen möglich

2) Analyse & Bewertung

Erstellung eines Kriterienkatalogs für die Positionierung der einzelnen Anwendungen im Portfolio, evtl. durch Expertenrunde mit Kreativtechnik, unstrukturierte Kriterien kategorisieren, Gewichtung der Kriterien, Messung der Ausprägungen, Ist-Zustand

3) Zielbestimmung

Entwicklungsbedarf und Potential der Anwendungen klären, Portfolio für den Soll-Zustand der Anwendungen

4) Handlungsbestimmung

Planung der Schritte, die zur Erreichung des Soll-Zustandes durchgeführt werden müssen, Interdependenzen berücksichtigen

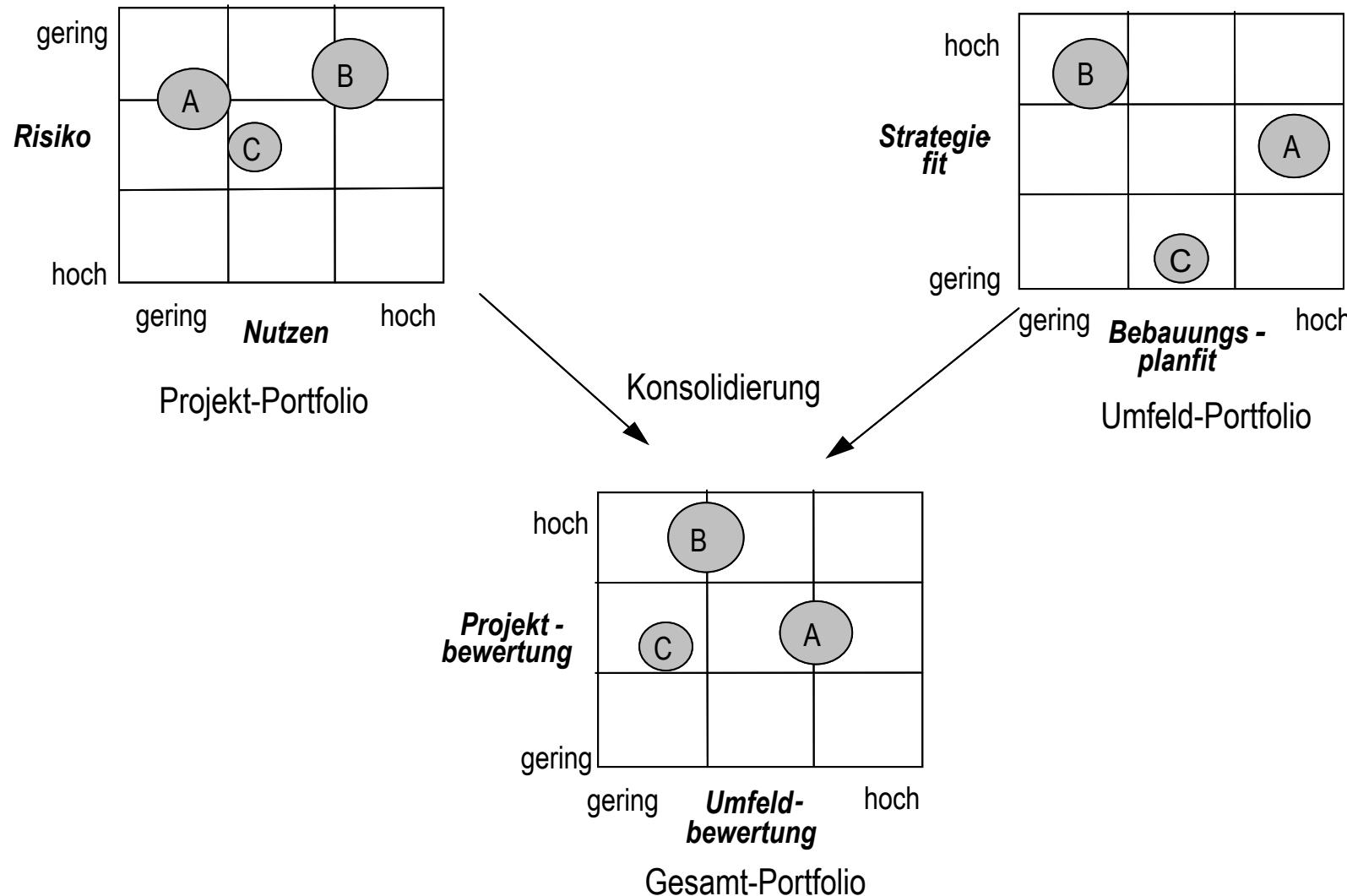
6.3 Gesamtheit der Anwendungen

Kriterien der Anwendungs-Portfolio-Bewertung

Risiko	Nutzen	Strategiefit	Bebauungsplanfit
Projektdauer	Wirtschaftlichkeit	Kundenorientierung	Prozessorganisation
Projektgröße	Nutzungsdauer	Konkurrenzorientierung	Prozessverantwortung
Ressourcenverfügbarkeit	Nicht quantifizierbarer Nutzen	Prozessorientierung	Prozessziele
Problemdimension	Mitarbeiterorientierung	Effizienz der Abwicklung	IS-Architektur-Daten
Anhängigkeit	Potentialentwicklung		IS-Architektur-Funktionen
			IT-Strategie-Technologiefit

6.3 Gesamtheit der Anwendungen

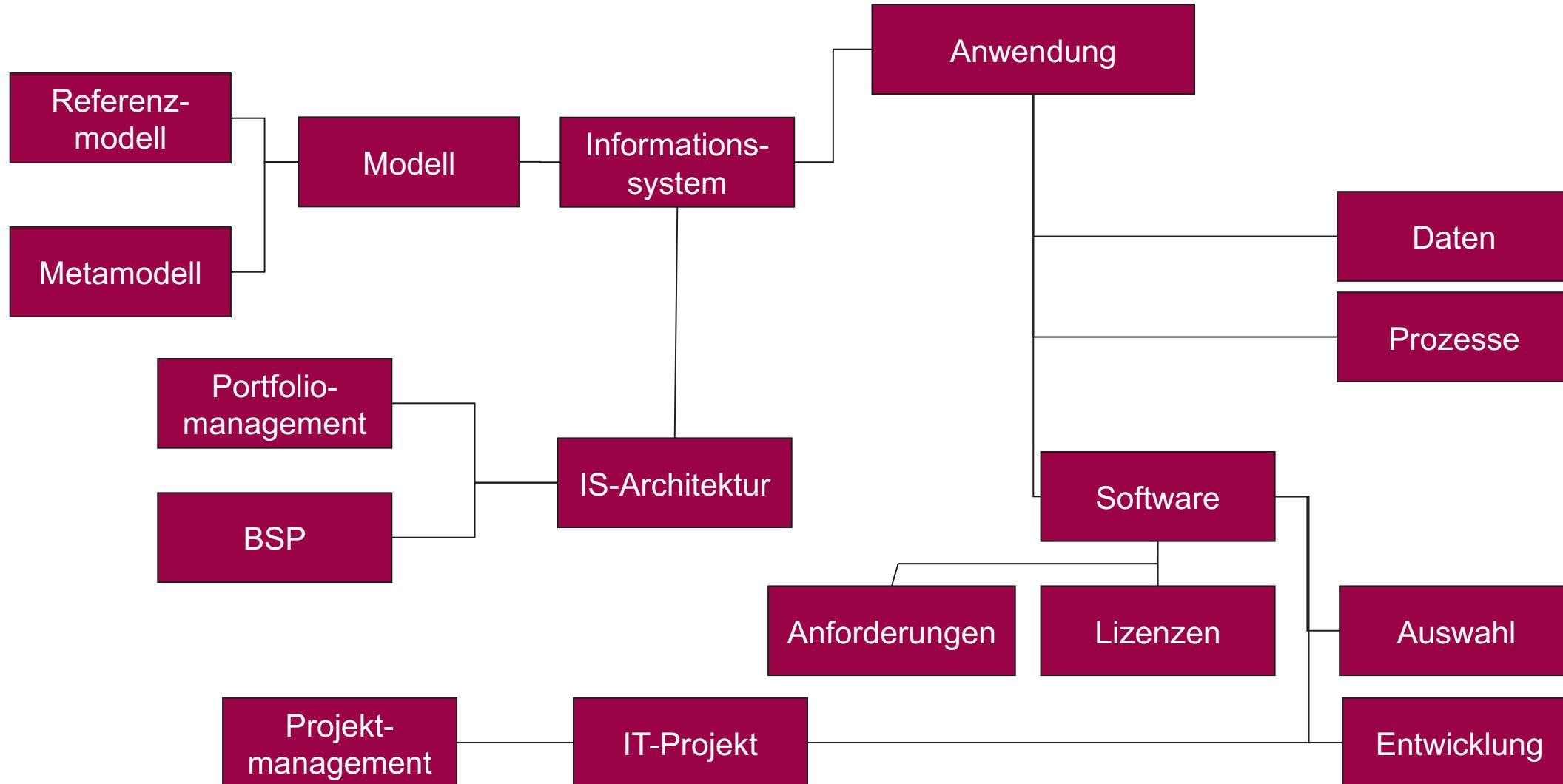
Dimensionen des IS-Projekt-Portfolios



Quelle: Krcmar (2015), Informationsmanagement, S. 296

6.3 Gesamtheit der Anwendungen

Begriffe zu LE 6



Und nun sind Sie dran...





Nach jeder Lerneinheit:

- Erstellen Sie Single Choice Aufgaben (Wahr/Falsch)

Ablauf:

- Sie brauchen nur Zettel und Stift
- Alle erstellen 3 Wahr-Falsch-Aussagen (Lernziele!)
- Mit Nachbarn tauschen und gegenseitig beantworten / diskutieren
- Zum Schluss abgeben (idealerweise physisch, notfalls Mail)

Ziel:

- Reflexion des theoretischen Inputs
- Anreicherung des Fundus an Klausuraufgaben

Lernziele LE06 – Management der Anwendungen



- 1) Sie kennen das Konzept des **Anwendungslbenszyklus** und die Aufgaben, die im Rahmen des Lebenszyklus anfallen.
- 2) Sie können Methoden zur **Softwareauswahl**, zur **Beurteilung von Lizenzmodellen** und **Softwareentwicklung** erläutern, beurteilen und anwenden.

Quellen

Kernliteratur

- Krcmar, H.:
Informationsmanagement (2015), S. 203-314

Vertiefungsliteratur

- Österle, H. (1990). Integrierte Standardsoftware: Entscheidungshilfen für den Einsatz von Softwarepaketen. In Österle, H. (Ed.), *Band 2: Auswahl, Einführung und Betrieb von Standardsoftware*. Halbergmoos.
- Brügge, B.; Harhoff, D.; Picot, A.; Creighton, O.; Fiedler, M.; Henkel, J. (2004). *Open-Source-Software: Eine ökonomische und technische Analyse*. Berlin, Heidelberg: Springer.
- Heinrich, L.J. (2002). *Informationsmanagement: Planung, Überwachung und Steuerung der Informationsinfrastruktur*. (7., vollständig überarbeitete und ergänzte Auflage). München, Wien: Oldenbourg. S. 263ff



Prof. Dr. Matthias Söllner

Fragen zur Vorlesung können Sie gerne via Mail an soellner@uni-kassel.de richten.

Weitere Informationen zum Fachgebiet finden Sie unter: www.uni-kassel.de/go/wise

