UNIVERSITÄT BERN

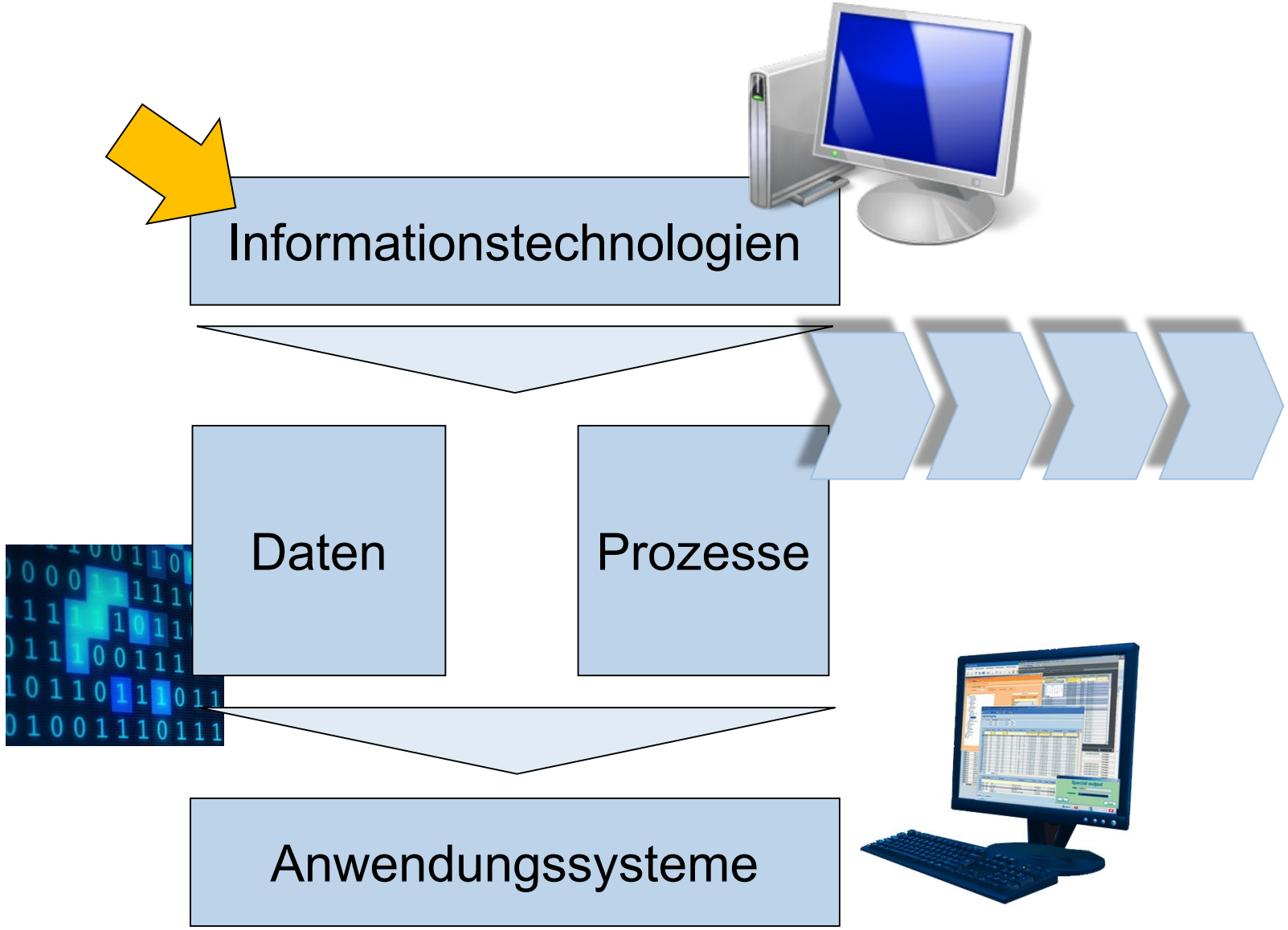
Einführung in die Wirtschaftsinformatik

Typen von Software

Prof. Dr. Thomas Myrach
Universität Bern
Institut für Wirtschaftsinformatik
Abteilung Informationsmanagement

Bern, 04. März 2020

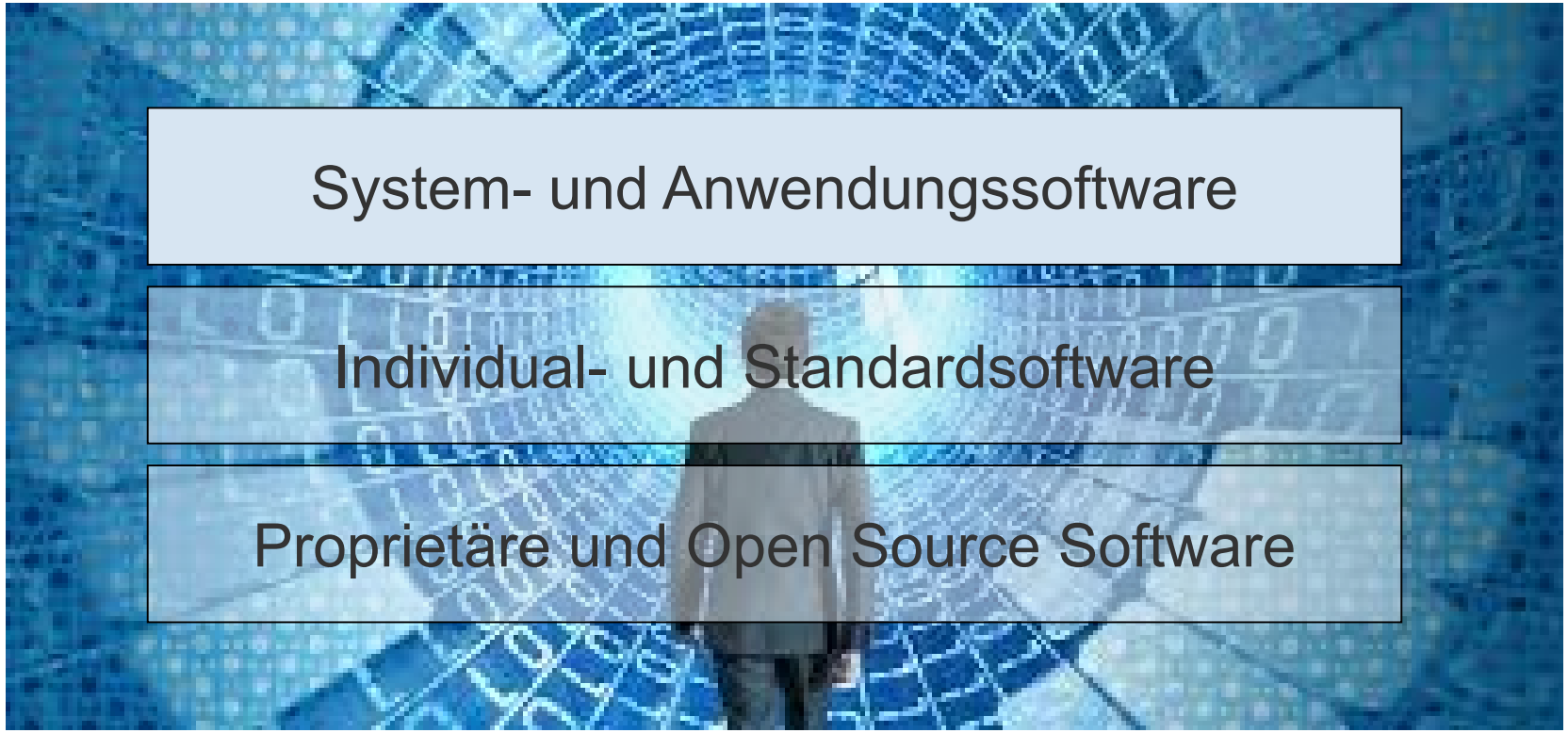
Logischer Aufbau



Lernziele

- Sie kennen den Unterschied von Systemsoftware und Anwendungssoftware.
- Sie wissen, welche Aufgaben ein Betriebssystem übernimmt.
- Sie können einen Überblick über drei verschiedene Typen betrieblicher Anwendungssysteme geben.
- Sie können Individualsoftware und Standardsoftware unterscheiden.
- Sie kennen die Argumente, die für und gegen Standardsoftware sprechen.
- Sie wissen, was Open Source Software ist und kennen die Bedeutung von Lizenzen bei der Nutzung von Software.
- Sie kennen das Konzept Total Cost of Ownership und welche Rolle es bei der Beurteilung von Open Source Software spielt.

Gliederung


System- und Anwendungssoftware
Individual- und Standardsoftware
Proprietäre und Open Source Software

Begriff Software

- Sammelbegriff für die immateriellen Komponenten eines Rechnersystems.
- Die Abgrenzung ist nicht ganz klar, was unter Software zu zählen ist.
- Im Zentrum steht Programmcode, der durch eine CPU zur Ausführung gebracht werden kann.
- Durch Software können mit Rechnern spezifische Aufgaben bearbeitet werden.
- Programmcode wird als digitales Artefakt in Dateien gespeichert.

Systemstapel

- Für die Arbeit mit einem Rechner braucht es Hardware und Software.
- Hardware und Software lassen sich nach Schichten unterscheiden.
- Bei Software wird mindestens unterschieden zwischen
  - Systemsoftware und
  - Anwendungssoftware.
- Da diese Schichten aufeinander aufbauen, wird von einem Systemstapel gesprochen.

Systemsoftware

- Systemsoftware wird abgegrenzt von Anwendungssoftware.
- Systemsoftware bietet eine Plattform für andere Software.
- Systemsoftware stellt eine Verbindung zur Hardware her und steuert die Verwendung dieser Ressourcen.
- Sie verwaltet sowohl die internen als auch die externen Hardwarekomponenten und kommuniziert dazu mit diesen.
- Zur Systemsoftware gehören vor allem Betriebssysteme
- Weiterhin wird auch systemnahe Software dazugerechnet.

Generelle Aufgaben eines Betriebssystems

- Den Anwendern sollen Abstraktionen der Betriebsmittel zur Verfügung gestellt werden.
  - Das Betriebssystem bietet eine leichter verständliche und besser handhabbare Schnittstelle zur eigentlichen Maschine an.
  - Es verdeckt die Komplexität der darunterliegenden Maschine.
  - Das Betriebssystem erzeugt abstrakte Objekte, um die Komplexität beherrschbar zu machen.
- Die Hardwareressourcen sollen verwaltet werden.
  - Das Betriebssystem ordnet und kontrolliert die Allokation der Prozessoren, Speicher und der Ein-/Ausgabegeräte.
  - Auf einem modernen Betriebssystem können mehrere Programme gleichzeitig ausgeführt werden.
  - Das Betriebssystem überwacht, welches Programm gerade welches Betriebsmittel nutzt.

Abstraktion der Betriebsmittel

Beispiel: Benutzerschnittstelle (Bildschirm, Eingabegeräte)

- Aufbau einer grafischen Benutzeroberfläche (GUI: *Graphical User Interface*) für die Interaktion mit dem Benutzer.
- Verwendet oftmals die sogenannte Schreibtischmetapher (Desktop).
- Zentrale Elemente des "Schreibtischs" sind Fenster und Piktogramme (Icons).
- Icons stellen vor allem (ausführbare) Programme und Dateien dar.
- Von der Metapher her liegen auf dem Desktop nicht nur geschlossene Dokumente (Dateien), sondern auch die (Fenster der in Programmen) geöffneten Dokumente.

Betriebssysteme

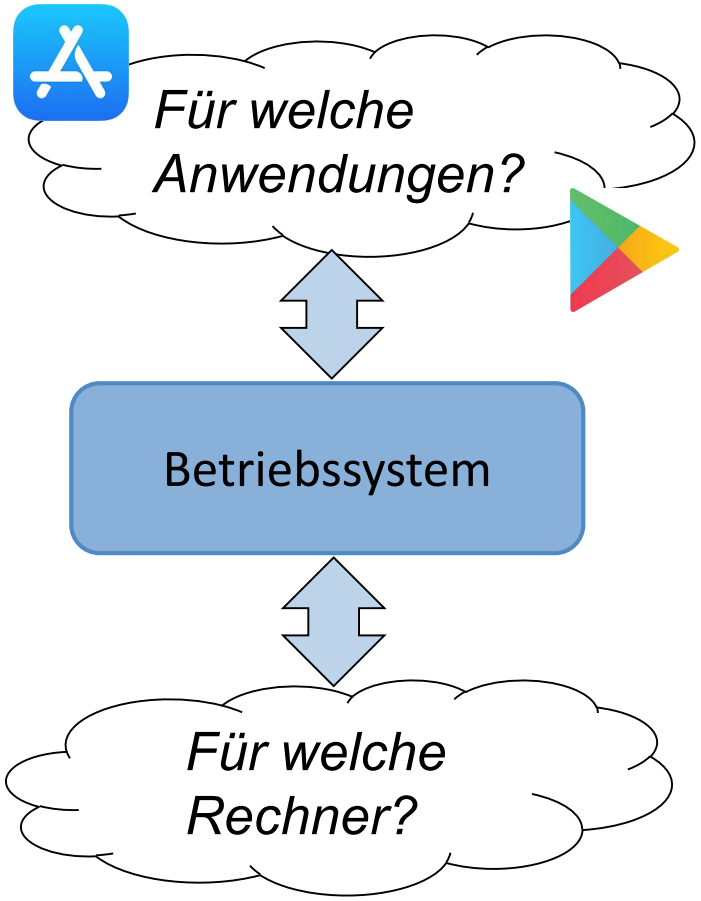
UNIVERSITÄT BERN

- Für gängige Endgeräte (Personal Computer, Tablets, Smartphones) sind wenige Betriebssysteme weit verbreitet.
- Bei Personal Computern:
  - Microsoft Windows
  - Apple OS X
  - Linux
- Bei Smartphones und Tablets:
  - Apple iOS
  - Android

	System	2019	2018	2017
1.	Android	39,09 %	38,97 %	31,76 %
2.	Windows	35,85 %	37,07 %	43,82 %
3.	iOS	13,98 %	13,18 %	11,71 %
4.	macOS	6,63 %	5,24 %	5,09 %
5.	Linux	0,80 %	0,76 %	0,94 %
	nicht identifizierte BS	2,36 %	2,72 %	3,64 %
	andere BS	1,29 %	2,06 %	3,03 %

Verbreitung anhand von Web-Zugriffen (StatCounter)

Einordnung Betriebssystem



- Betriebssysteme haben eine Mittlerfunktion zwischen Hardware und Anwendungssoftware.
- Betriebssysteme sind auf bestimmte Rechner zugeschnitten.
- Anwendungssoftware ist an bestimmte Betriebssysteme gebunden.
- An ein Betriebssystem sind Software-Ökosysteme geknüpft.
- Das Software-Ökosystem ist ein wichtiges Argument bei der Wahl eines Betriebssystems als Plattform.

Anwendungsoftware

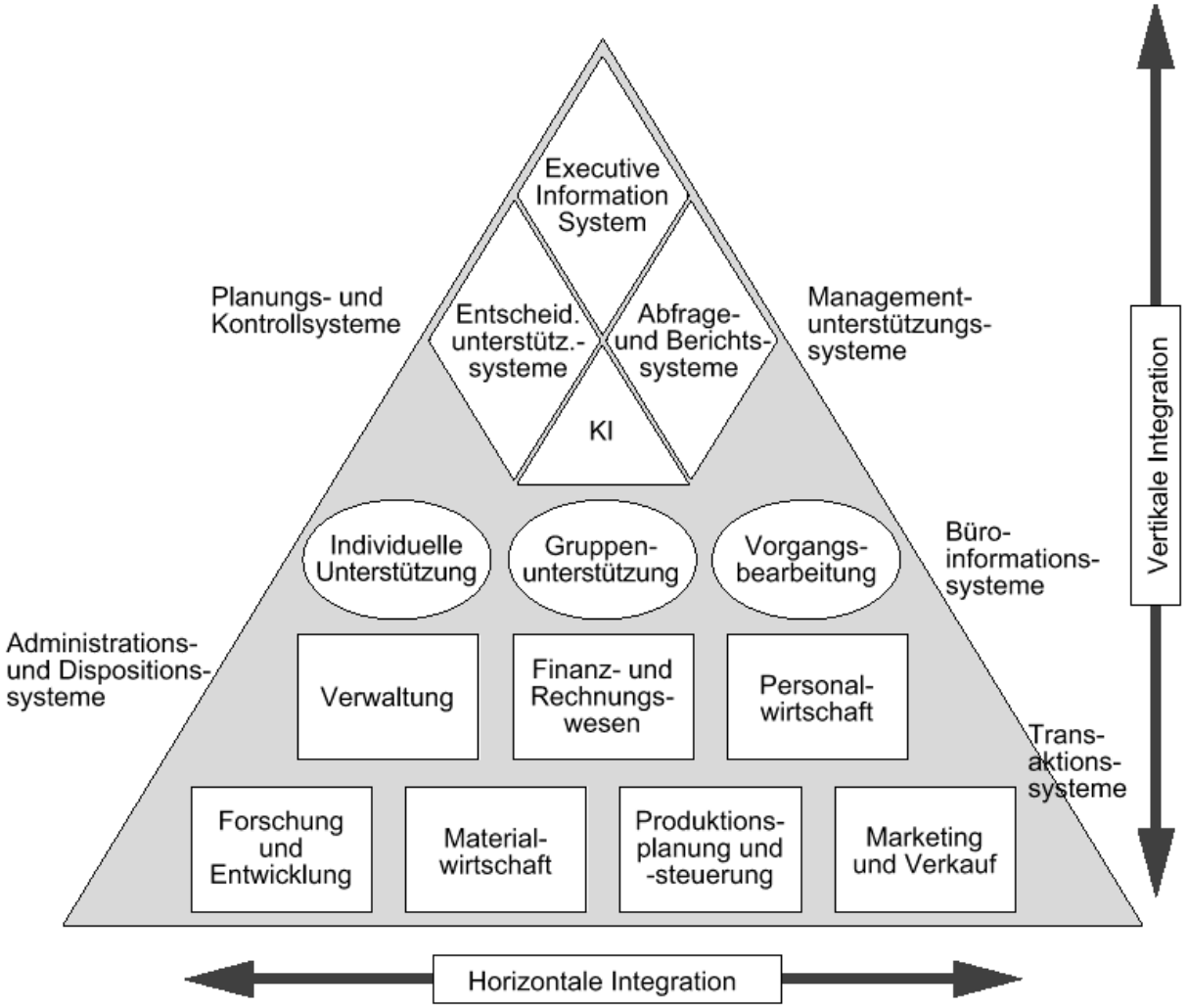
UNIVERSITÄT BERN

- Eine Anwendung von Informations- und Kommunikationstechnologien auf (betriebliche) Aufgabenstellungen.
- Anwendungssysteme sollen die Benutzer beim Durchführen der Geschäftsprozesse und betrieblichen Funktionen effektiv unterstützen.
- Entsprechend den sehr unterschiedlichen betrieblichen Aufgaben werden verschiedenartige Anwendungen eingesetzt.
- Anwendungssysteme werden oftmals als Standardsoftware-Pakete auf dem Markt gekauft.

Vielfalt von Anwendungssystemen

- In der betrieblichen Praxis wird eine Vielzahl unterschiedlicher Anwendungssysteme eingesetzt.
- Diese lassen sich durch eine Reihe von Charakteristika voneinander abgrenzen:
  - Generell anwendbare Werkzeuge versus Werkzeuge für eine spezifische Problemstellung.
  - Systeme für operative Aufgaben versus Systeme zur Unterstützung von Führungsfunktionen.
  - Systeme für individuelle Aufgaben versus Systeme für personen- und bereichsübergreifende Aufgabenstellungen.
  - …
- Durch Typologien soll die Vielfalt der Werkzeuge systematisiert werden.

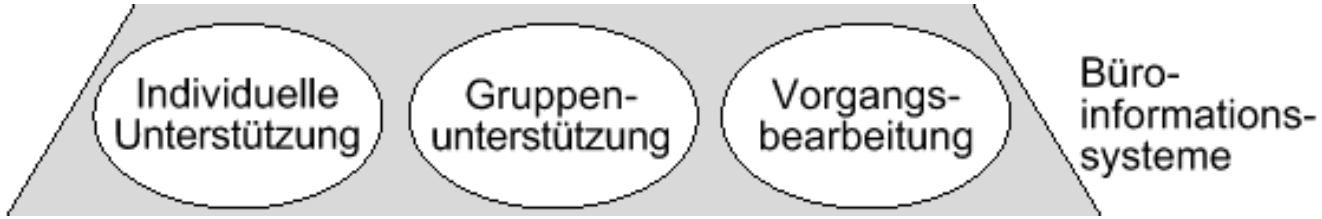
Typen von Anwendungssystemen



Büroinformationssysteme

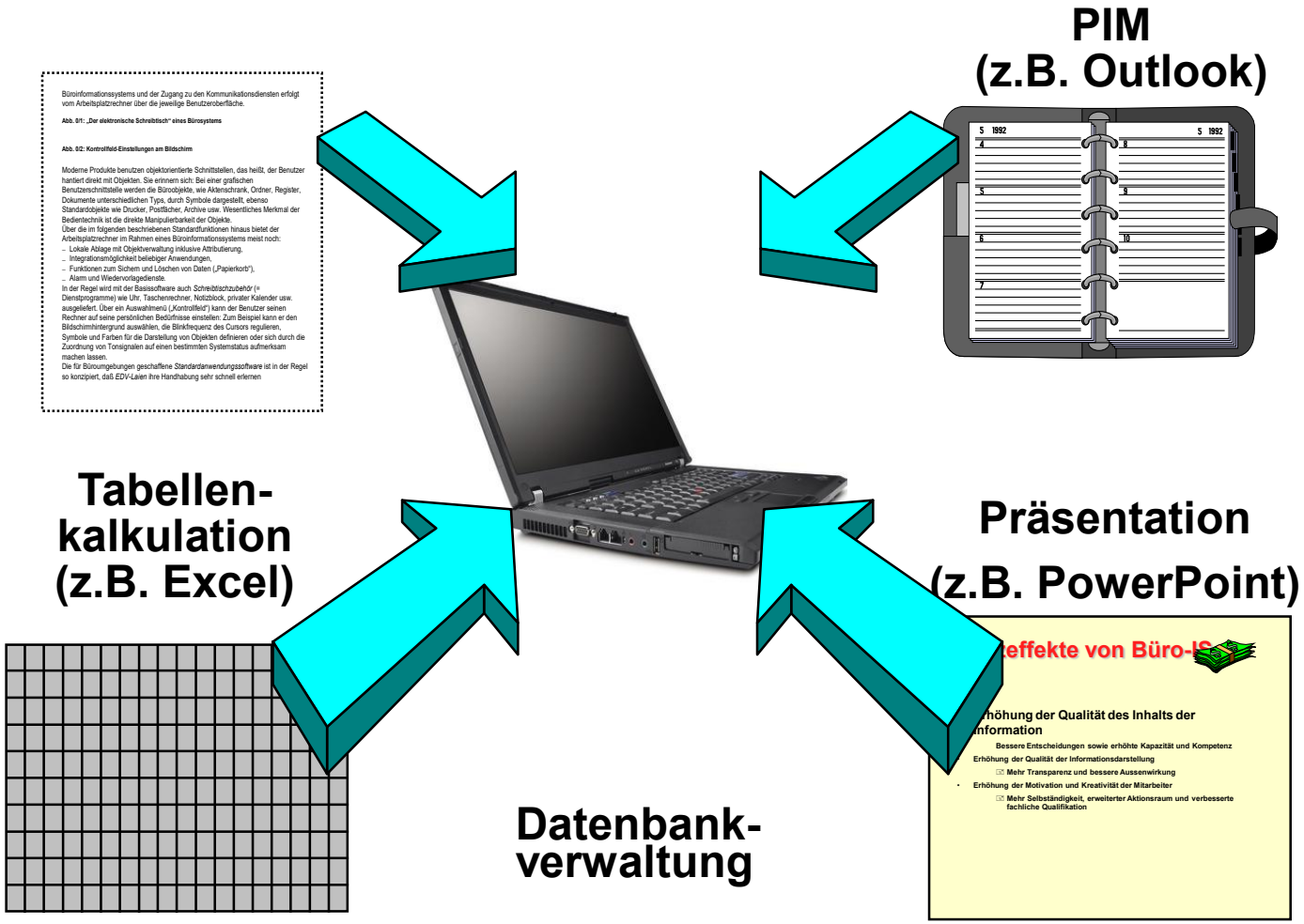
Überblick

- Büroinformationssysteme unterstützen typische Büroarbeit, wie sie bei administrativen Vorgängen anfällt.
- Es handelt sich vor allem um Werkzeuge der individuellen Unterstützung von Tätigkeiten der Büroarbeit.
- Daneben sind jedoch auch Instrumente für die Unterstützung der Arbeit in Gruppen relevant.
- Diese weisen kollaborative Funktionen auf, wie das Unterstützen des Austausches von Dokumenten.



Büroinformationssysteme

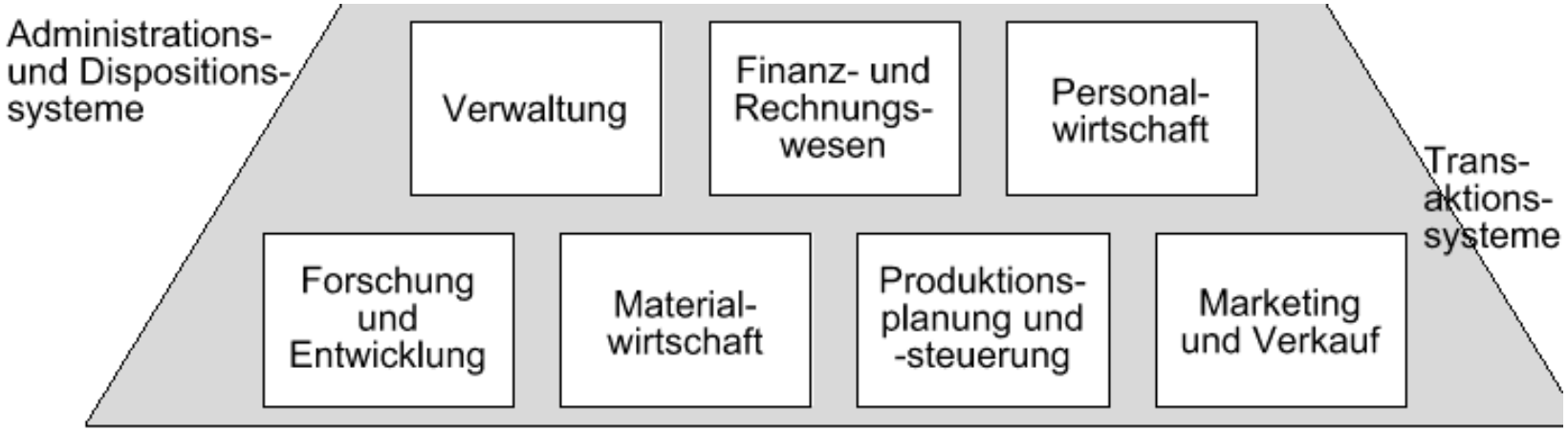
Beispiel: Integrierte Endbenutzerwerkzeuge



Unterstützung betrieblicher Leistungsprozesse

Überblick

- Die betrieblichen Leistungsprozesse werden mittlerweile weitgehend von IS unterstützt.
- Historisch sind Insellösungen für die verschiedenen betrieblichen Funktionsbereiche entstanden.
- Diese wurden mittlerweile vielfach durch integrierte Standardlösungen abgelöst.



Unterstützung betrieblicher Leistungsprozesse

Beispiel: ERP (Enterprise Resource Planning)

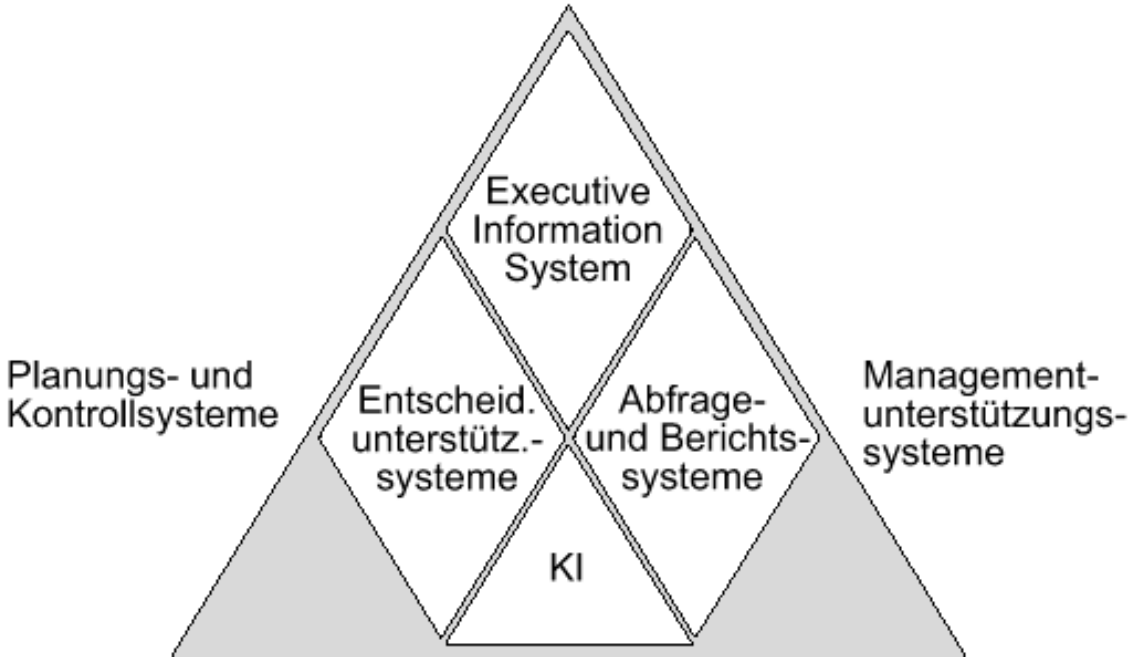
- A **collection of applications**
- that can be used to **manage the whole business**
- which integrate
  - sales,
  - manufacturing,
  - human resources,
  - logistics,
  - accounting,
  - and other enterprise functions,
- and allow all functions to **share a common database** and business analysis tools.

[Gartner Group, ca. 1990]

Managementunterstützung

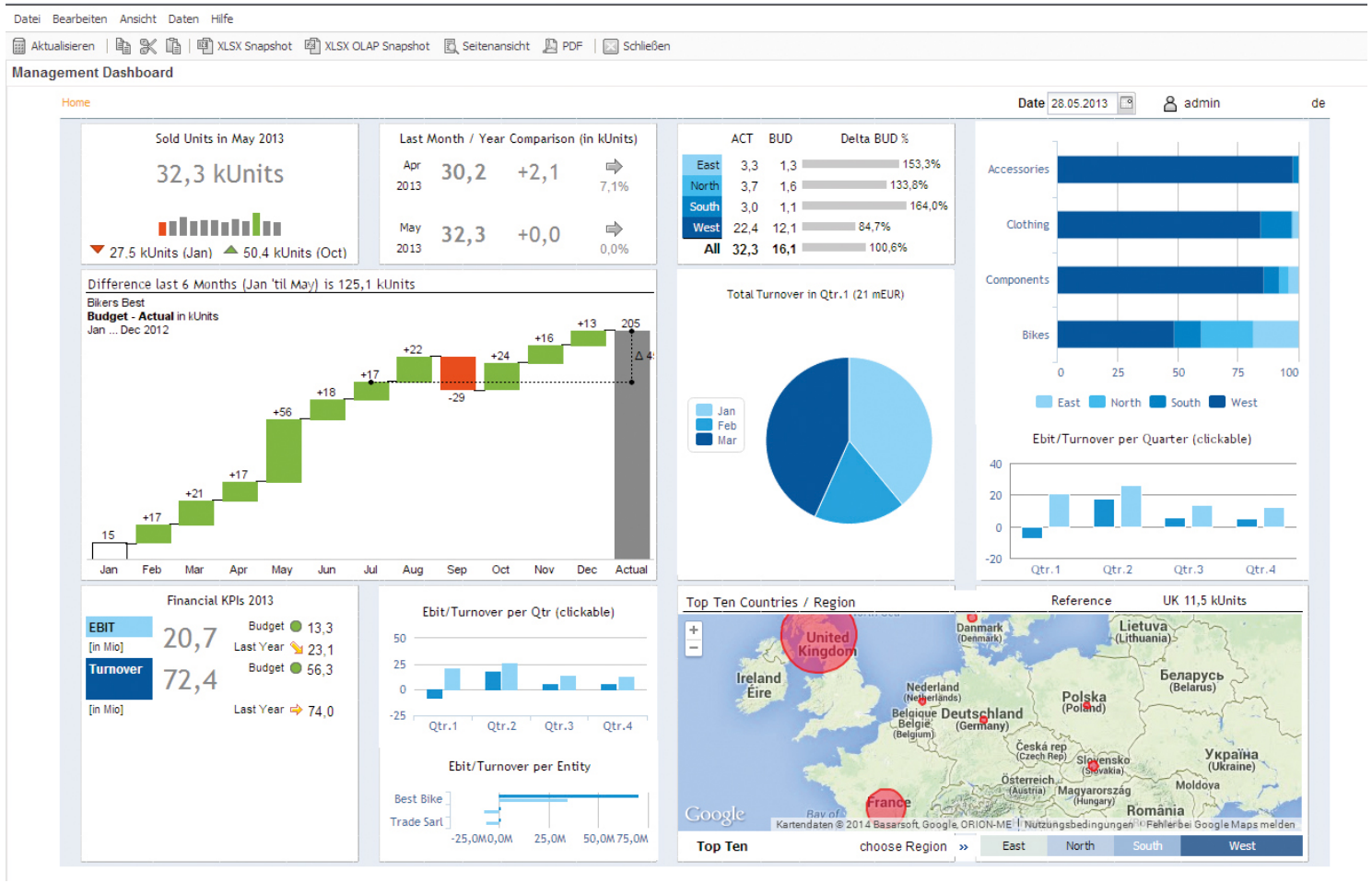
Überblick

- Managementunterstützungssysteme sollen Führungskräften eine adäquate Informationsversorgung und Entscheidungsunterstützung bieten.



Managementunterstützung

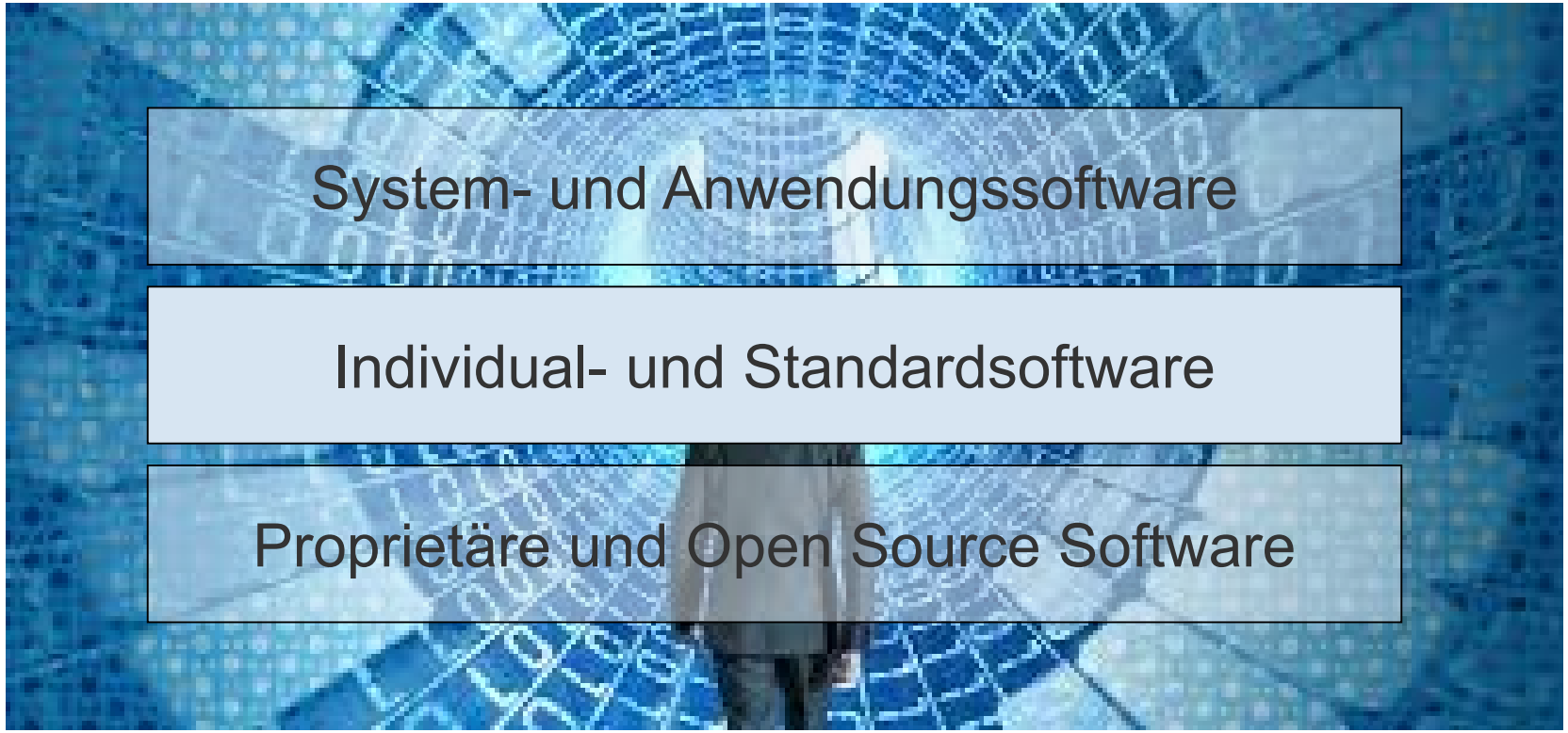
Beispiel: Management-Cockpits (Dashboards)



Systemintegration

- Daten müssen im Zuge der Bearbeitung von Prozessen zwischen den jeweiligen Handlungsträgern fliessen.
- Die Verfügbarkeit der Daten für alle sie benötigenden Aufgaben bedingt eine Integration der IS.
- Horizontale Integration
  - Integration zwischen Systemen der gleichen Ebene.
  - Beispiel: Integration zwischen Modulen eines ERP-Systems.
  - Beispiel: Integration zwischen Office-Werkzeugen.
- Vertikale Integration
  - Integration zwischen Systemen unterschiedlicher Ebenen.
  - Beispiel: Integration eines ERP-Systems mit einem Data Warehouse.

Gliederung


System- und Anwendungssoftware
Individual- und Standardsoftware
Proprietäre und Open Source Software

Anforderungen und Anwendungssoftware

- Anwendungssoftware muss bestimmten funktionalen und nicht-funktionalen Anforderungen entsprechen.
- Anforderungen können sehr spezifisch für den jeweiligen Betrieb oder die zu unterstützende Aufgabe sein.
- In diesem Fall kann ein Anbieter beauftragt werden eine **individuelle Software** zu entwickeln.
- Anforderungen können aber auch allgemeiner und für unterschiedliche Betriebe in ähnlicher Form relevant sein.
- In diesem Fall kann ein Anbieter auf die Idee kommen, eine **standardisierte Software** für dieses allgemeine Bedürfnis zu entwickeln und dann zu verkaufen.

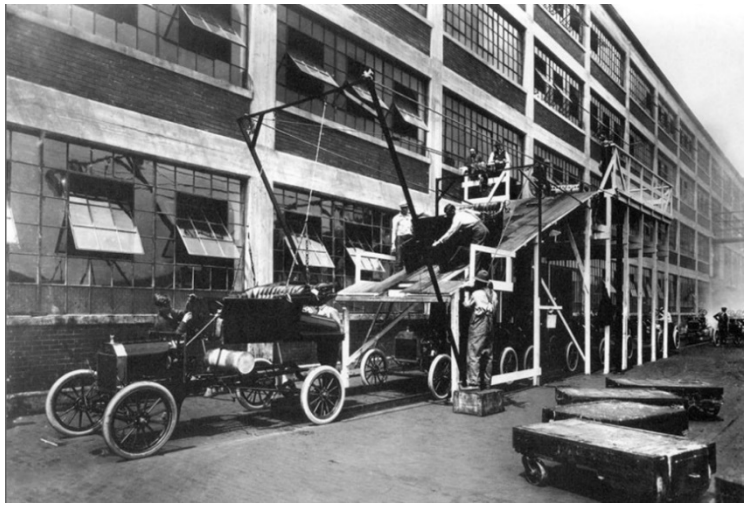
Individual-Software

- Individual-Software wird spezifisch für einen bestimmten Anwender und seine Bedürfnisse entwickelt.
- **Vor- und Nachteile**
  - Vorteil: Entspricht genau den Anforderungen der Anwender.
  - Nachteil: Entwicklungsaufwand wegen „Massarbeit“ relativ hoch.
- **Warum Individual-Software?**
  - Keine passende Standard-Software vorhanden
  - Anpassung der Standard-Software käme teurer als Neuentwicklung von Individual-Software
  - Wettbewerbsvorteil durch individuelle Entwicklung
  - Abhängigkeit von externen Anbietern reduzieren (wenn im Besitz vom Urheberrecht)

Standard-Software

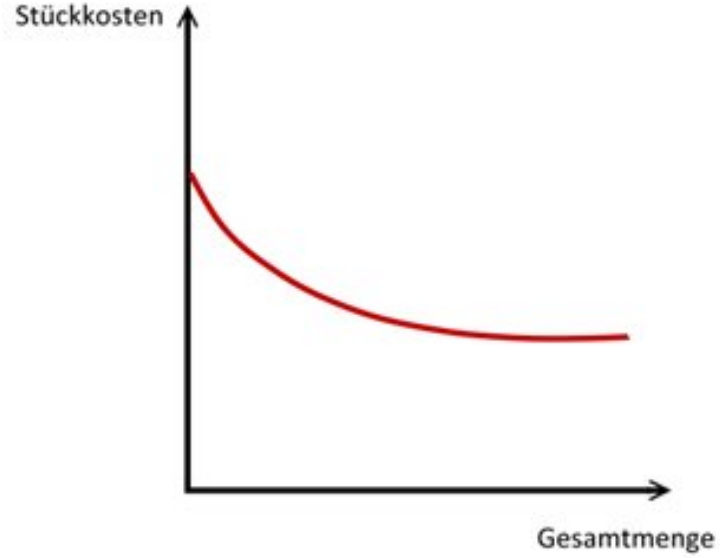
- Standard-Software wird zur Erfüllung bestimmter funktionaler Anforderungen für einen breiten Kreis von Anwendern entwickelt.
- **Vor- und Nachteile**
  - Vorteil: Kosten für Erst-Entwicklung und Weiterentwicklung werden auf mehrere Anwender verteilt.
  - Nachteil: Anforderungen der einzelnen Nutzer werden nicht genau abgedeckt.
- **Warum Standard-Software?**
  - Ist günstiger falls Anforderungen ausreichend abgedeckt werden
  - Profitieren von Weiterentwicklungen für andere Anwender
  - Kann rascher eingeführt werden weil Software bereits besteht
  - Weniger Risiko da bestehende Software getestet werden kann

Ökonomische Logik der Massenfertigung



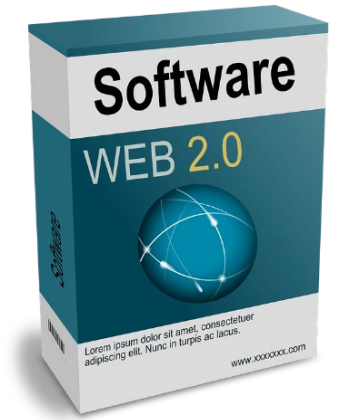
- Bei einem materiellem Gut entstehen Kosten bei der Entwicklung und der Fertigung.
- Die Entwicklungskosten fallen nur einmal an.
- Sie werden bei der Fertigung auf jedes Stück verteilt.

- Die Massenfertigung standardisierter Güter ist ein wesentliches Merkmal der Industrialisierung.
- Die Produktion in grösseren Mengen führt zu Kostenvorteilen (*Economies of Scale*).



Skaleneffekte bei Software

- Software ist ein immaterielles Gut.
- Software manifestiert sich in (digitalen) Computerdateien.
- Der Hauptteil der Kosten entsteht in der Entwicklung.
- Die Entwicklungskosten lassen sich über eine Fixkostendegression auf die verschiedenen Nutzer verteilen.
- Computerdateien können ohne grossen Aufwand kopiert und verbreitet werden.
- Dieses führt zu ungleich höheren Kostenvorteilen als bei materiellen Gütern.

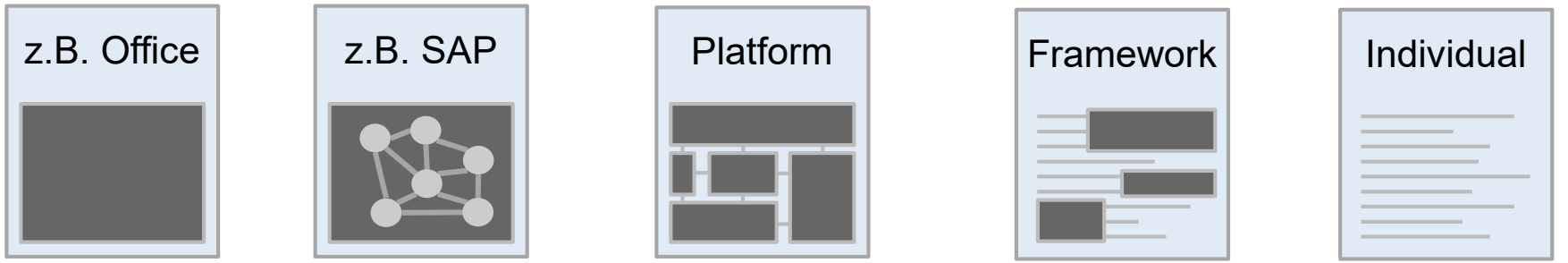


Anpassung als Grundproblem

- Standardsoftware hat den Nachteil, dass sie nicht auf spezifische Anwenderbedürfnisse hin konzipiert ist.
- Dies ist bei allgemeinen Problemstellungen relativ unerheblich.
- Bei spezifischen Problemstellungen, z.B. der Unterstützung betrieblicher Prozesse, kann dies jedoch problematisch sein.
- Um besondere betriebliche Gegebenheiten abzubilden braucht es unter Umständen Anpassungen.
- Standardsoftware bietet häufig eine Reihe von internen Einstellungsmöglichkeiten, um benutzer- oder betriebs-spezifische Anpassungen vorzunehmen (Customizing).

Kontinuum von Standard-Software bis Individual-Software

Standard-Software **Individual-Software**



Installieren	Konfigurieren, parametrisieren	Integration von bestehenden Komponenten	Verwendung bestehendes Programmier-Framework	Alles neu programmieren
buy	***customize***	***assemble***	***build***	***make***

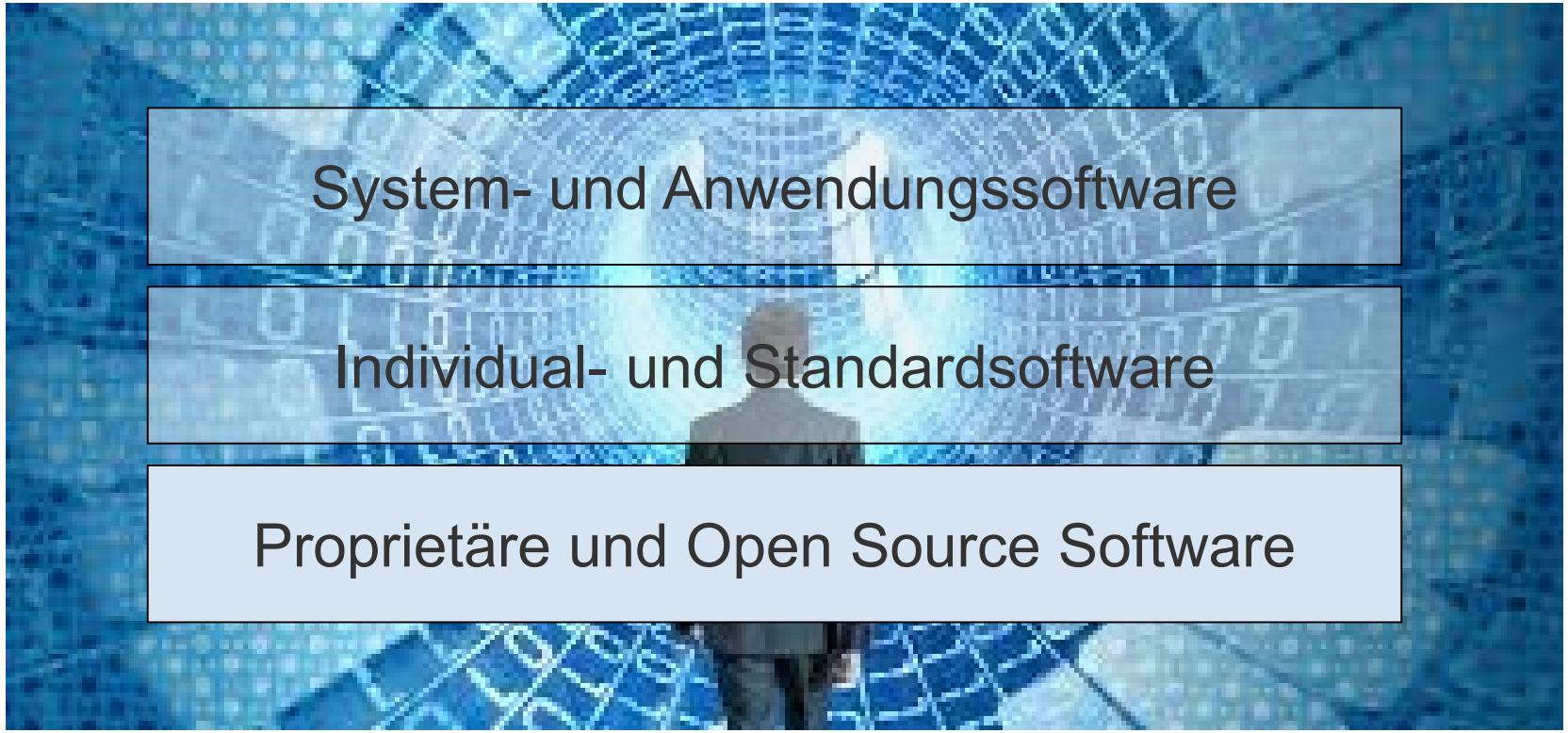
nach M. Stürmer

Fazit

UNIVERSITÄT BERN

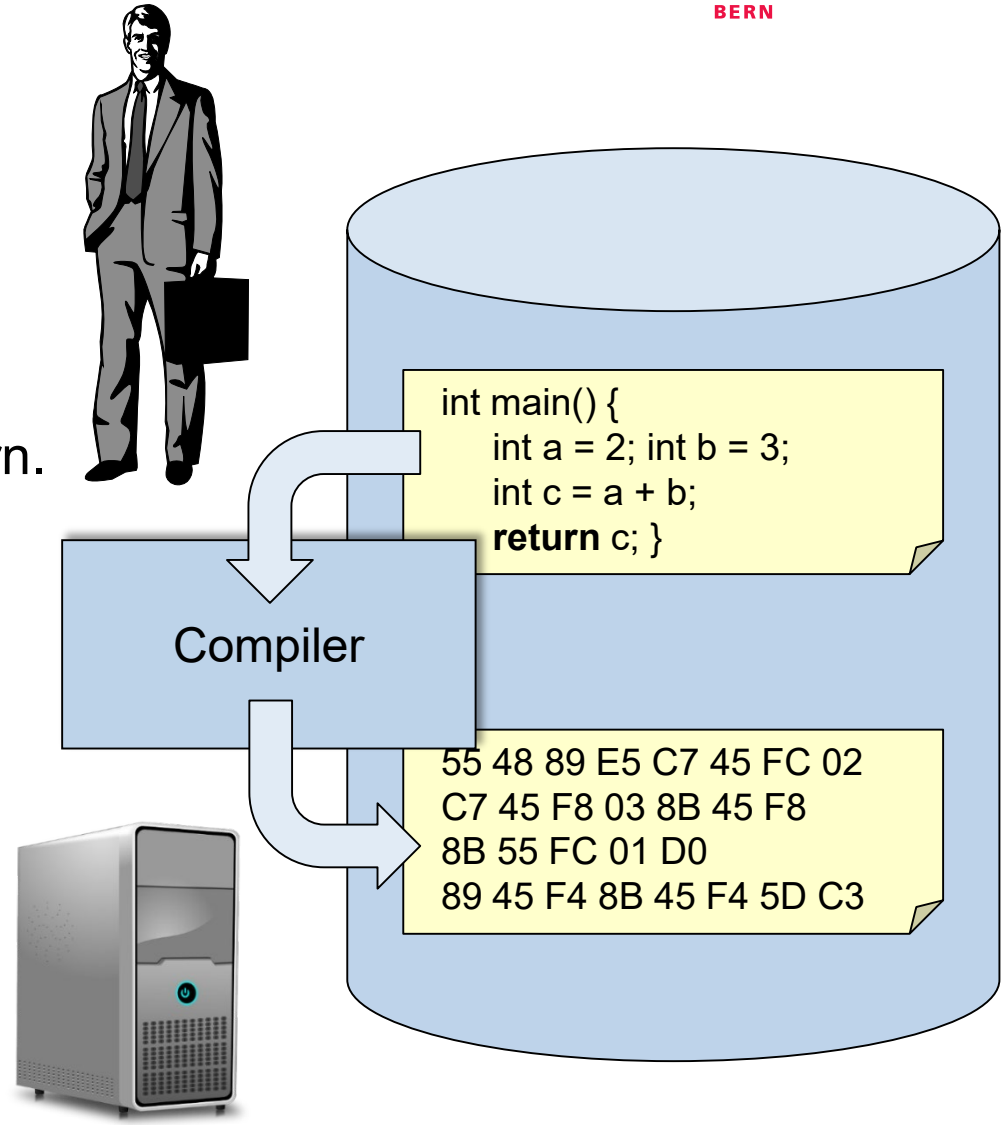
- Standard-Software wird noch generellen Anforderungen für einen bestimmten Kundenkreis entwickelt.
- Damit sie möglichst breit anwendbar sind, sollten sie unterschiedliche Anforderungen abdecken können.
- Standardisierung von Software ist wegen der grossen Skaleneffekte ökonomisch sehr sinnvoll.
- Zentrales Problem ist die Anpassung auf spezifische betriebliche Anforderungen.
- Individual- und Standardsoftware sind keine dichotomen Konzepte, sondern bilden ein Kontinuum mit unterschiedlichen Ausprägungen.

Gliederung


System- und Anwendungssoftware
Individual- und Standardsoftware
Proprietäre und Open Source Software

Arten von Programmdateien

- **Quellcode**
  - Im Syntax einer (höheren) Programmiersprache.
  - Abstraktion richten sich an Menschen.
  - Menschen können Quellcode lesen und verändern.
  - Kann von der Maschine nicht verarbeitet werden.
- Ausführbarer **Maschinencode**
  - In der Maschinensprache eines Prozessors.
  - Code kann von der Maschine verarbeitet werden.
  - Menschen können Maschinencode nicht einfach lesen und verändern.

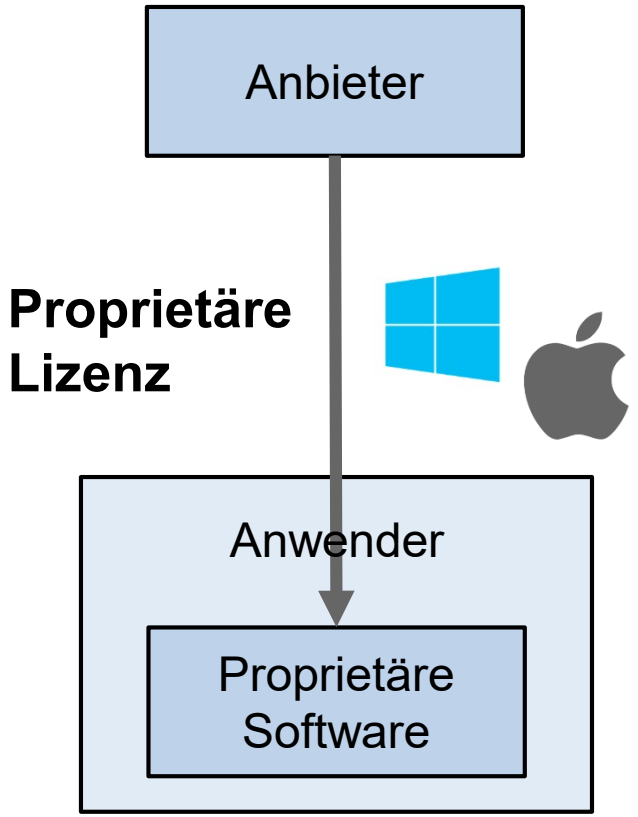


Software und Lizenzen

- Nutzung
  - Lizenzen erlauben die Nutzung der Software durch den Lizenznehmer.
  - Bezüglich der Art der Nutzung können Einschränkungen definiert werden.
  - Beispiele sind die Anzahl der Installationen bzw. der Nutzer.
- Weitergabe
  - Lizenzen schliessen die Weitergabe der Software durch den Lizenznehmer typischerweise aus.
  - Sicherung durch Programmschutz möglich, z.B. durch Lizenzschlüssel.
- Veränderung
  - Veränderungen sind realistischerweise nur im Quell-Code möglich.
  - Ist nur relevant, wenn der Quell-Code offengelegt wird (Open Source).

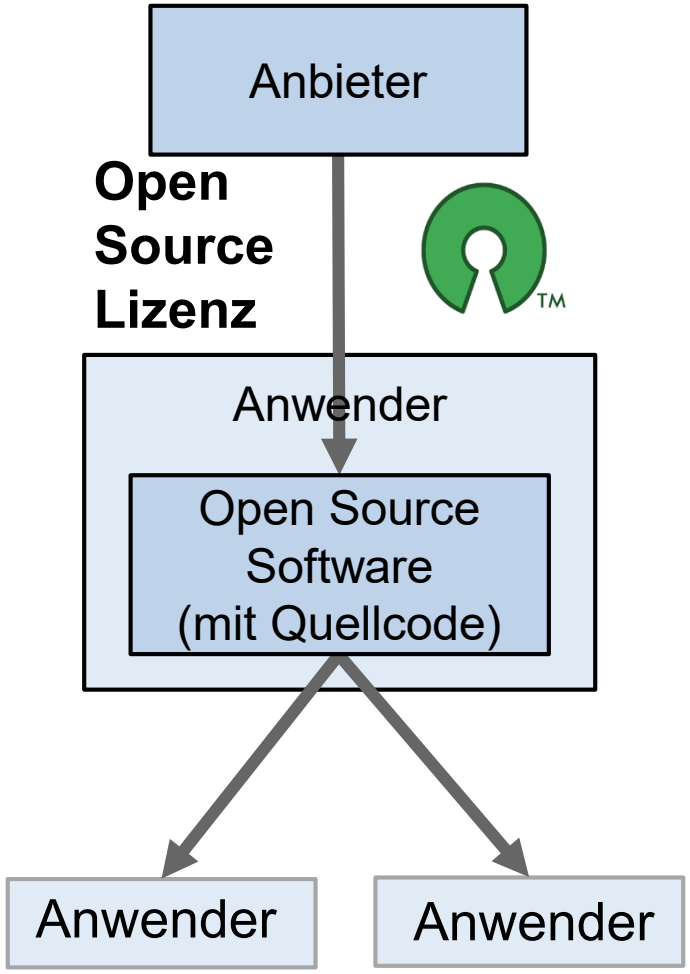
Proprietäre Software

- Anbieter entwickelt Software und verkauft **Nutzungslizenz an Anwender.**
- Der Quell-Code ist das Eigentum des Software-Erstellers und wird nicht offengelegt.
- Meist werden Dateien in ausführbaren Maschinencode ausgeliefert.
- Unter Umständen existiert ein Kopierschutz oder für die Aktivierung ist ein Lizenzschlüssel erforderlich.
- Anwender kann die Software gemäss **Lizenzbestimmungen** nutzen.
- Anwender darf Software nicht an andere weitergeben.



Open-Source-Software

- Anbieter ist häufig eine Community, die diese Software entwickelt.
- Der Quellcode einer Anwendung wird offen zur Verfügung gestellt.
- Vielfach ist es möglich, die Software ohne Kosten zu benutzen.
- Auch die Nutzung von Open-Source-Software ist an bestimmte **Lizenzen** gebunden.
- Diese regelt, was ein Benutzer mit der Software machen darf.



Open Source Lizenzen

Die **Open Source Initiative** (OSI) anerkennt rund 70 Lizenzen .

Alle Open Source Lizenzen geben vor:

1. Quellcode der Software ist zugänglich
2. Software darf beliebig eingesetzt werden
3. Software darf beliebig kopiert und verbreitet werden
4. Software darf verändert und der veränderter Form weitergegeben werden

Open Source Betriebssystem

Beispiel: Linux-Kernel

- Linux ist die bekannte Basis für Open-Source Betriebssysteme.
- Geeignet für unterschiedliche Hardware (PC, Server, Mobile, Embedded).
- Eigentlich handelt es sich bei Linux nur um den Betriebssystem-Kern (**Linux-Kernel**).
- Für die Nutzung als Betriebssystem wird der Linux-Kernel mit anderen freien Software-Komponenten ergänzt.
- Die weite Verbreitung erfolgte ab 1992 durch die Lizenzierung des Linux-Kernels unter der freien Lizenz **GPL.**
- Das Linux-Projekt ist verknüpft mit der Person **Linus Torvald.**

Open Source Betriebssystem

Beispiel: Ubuntu

- **Linux-basiertes** Betriebssystem für Endanwender, Schulen, Behörden, Firmen etc.
- Freie Alternative zu **Microsoft Windows**
- Läuft auf **Desktops (auch Mac), Tablets, Smartphones, Servers etc.**
- **Alle 6 Monate neue Version:** 18.04 aktuelle Version
- Software wird als Debian-Packages **übers Internet** installiert
- Wird von **vielen Millionen** Anwendern genutzt
- **Download** unter www.ubuntu.com

Open Source Bürosysteme

Beispiel: LibreOffice

- **Office-Suite** mit Text, Tabellen, Folienpräsentation, Grafik etc.
- Freie Alternative zu **Microsoft Office**
- Läuft auf **Windows, Mac und Linux**
- Auch als **Cloud** verfügbar: Collabora LibreOffice Online
- Unterstützt das offene **Dokumentenformat ODF**
- Wird von vielen **Firmen, Verwaltungen, Schulen, Privatpersonen etc.** verwendet
- **Download** unter www.libreoffice.org

Wirtschaftlichkeit von Open-Source-Software

- Ökonomisch gesehen erscheint es vernünftig, eine vergleichbare Software zu nutzen, die praktisch umsonst zur Verfügung steht.
- Allerdings sind einige Dinge zu beachten:
  - Bei der Nutzung von Software gelten sehr stark Netzeffekte.
  - Dies gilt insbesondere, wenn damit erzeugte Daten/Dokumente weitergegeben werden sollen.
  - Weiterhin können die TCO (Total Cost of Ownership) unter Umständen höher sein, als bei kommerziellen Standardprodukten.
  - Das Vorsichtsmotiv spricht häufig gegen Open-Source.

Total Cost of Ownership (TCO)

- Konzept der Gartner Group
- Ganzheitliche Erfassung der Informatikkosten.
  - Berücksichtigt werden alle Kosten, die in Zusammenhang mit der Anschaffung und dem Betrieb (inklusive der Wartung und Benutzerbetreuung) eines Informationssystems entstehen.
- TCO zielen auf die Vergleichbarkeit unterschiedlicher IT-Produkte.
  - PC versus Mainframe
  - **Proprietäre Software versus Open Source**
  - Inhouse-IT versus Outsourcing
- Kernaussage:
  - Nicht immer ist die vordergründig kostengünstigere Option bei ganzheitlicher Betrachtung auch tatsächlich die günstigere!

TCO-Kostenmodell: Kostenarten

Software-Kosten sind bei OSS typischerweise günstiger!

TCO = Direkte Kosten + indirekte Kosten

- Direkte Kosten

Hard- und Software
Abschreibungen, Leasinggebühren

Operations
Technischer Support, Informatikberufe

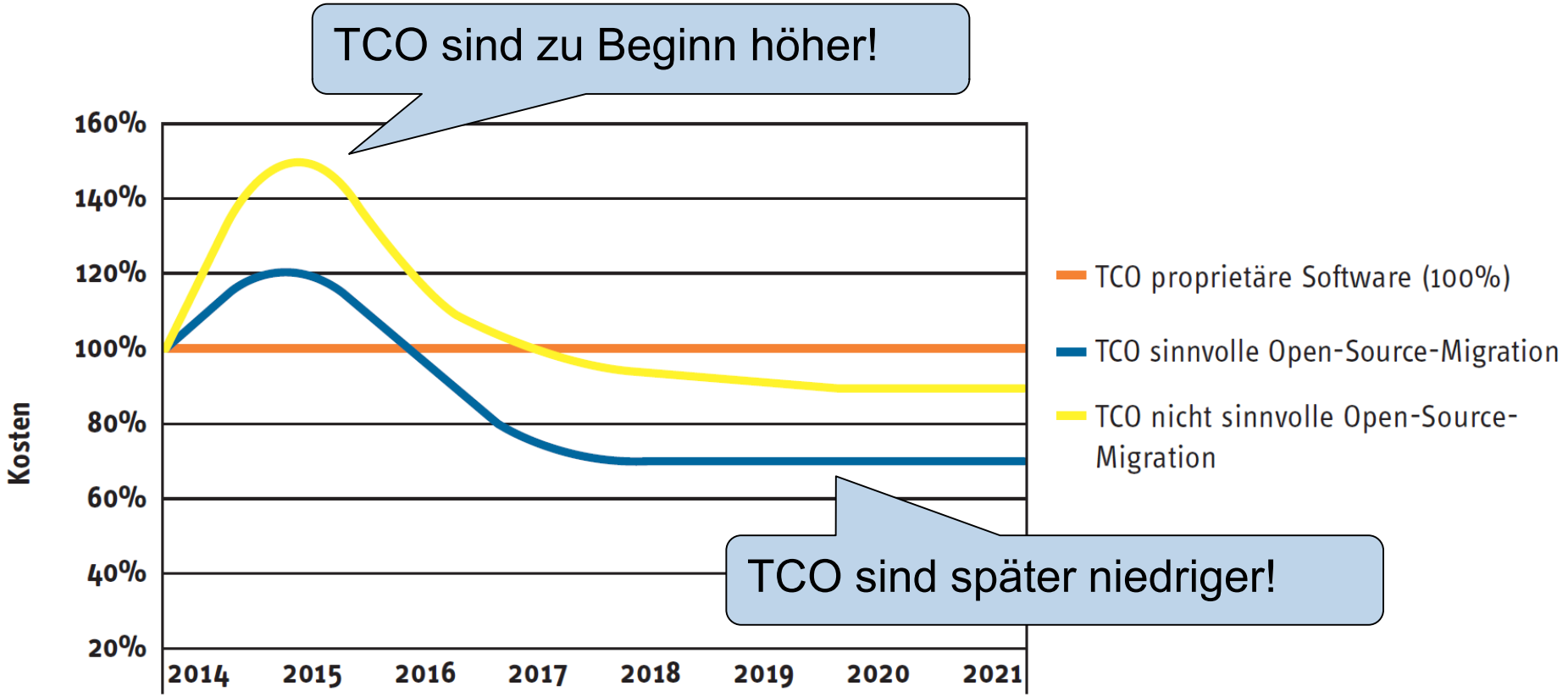
Verwaltung
EDV-Abteilung, Schulungsmassnahmen

- Indirekte Kosten -

End-User-Operations
Ausfallzeit der Endanwender durch Schulungsmassnahmen und Wahrnehmung von EDV Aufgaben

Downtime
Ausfallzeit der Endanwender durch (un)geplanten Ausfall der IT-Infrastruktur

Migration auf Open Source Produkte



Quelle: http://www.itmagazine.ch/Artikel/55166/Mehr_Geld__im_Portemonnaie__und_weniger__Sorgen_im_Gepaeck.html

Fazit

- Open-Source-Software beruht auf anderen Geschäftsmodellen als kommerzielle Software.
- Dies drückt sich auch in unterschiedlichen Lizenzmodellen aus.
- Weil der Quellcode frei verfügbar ist, kann er nicht gegenüber Veränderungen und Anpassungen geschützt werden.
- Dies wird in Open-Source-Lizenzmodellen berücksichtigt.
- Open-Source-Software wird oftmals kostenlos abgegeben.
- Trotzdem ist der Betrieb von Open-Source-Software nicht zwangsläufig günstiger als der von kommerzieller Software.
- Die ganzheitlichen Kosten des Software-Betriebs werden im Konzept der Total-Cost-of-Ownership (TCO) abgebildet.