

IT-Management

9. Standardsoftware

Prof. Dr. Thomas Allweyer

Standardsoftware

1. Softwareauswahl

1. Projektvorbereitung
2. Analyse
3. Anforderungen
4. Vorauswahl
5. Endauswahl

2. Einführung von Standardsoftware

3. Software-Lizenzmanagement

1. Lizenzformen
2. Lizenzmodelle
3. Beispiel: Microsoft
4. Beispiel: indirekte Benutzer
5. Beispiele im Outsourcing und Cloud-Computing
6. Folgen fehlerhafter Lizenzierungen
7. Ziele und Aufgaben des Lizenzmanagements

Standardsoftware

- **Individualsoftware**
 - Nur für spezielle, individuelle Anforderungen
 - Umsetzung neuer, innovativer Geschäftsmodelle
 - Optimal individuell angepasste IT-Unterstützung für Kernprozesse, die Wettbewerbsvorteile bringen

- **Standardsoftware**
 - Für Standardaufgaben und –prozesse lohnt sich Eigenentwicklung nicht.
 - Standardsoftware unterstützt typische Aufgaben meist besser.
 - Eigenentwicklung meist teurer
 - Kann angepasst werden.
 - Eigene Erweiterungen könnten hinzuprogrammiert werden.

- **Typische Beispiele für betriebswirtschaftliche Standardsoftware**
 - Enterprise-Resource-Planning (ERP)
 - Integrierte Unterstützung alle wesentlichen betrieblichen Funktionen
 - Customer-Relationship-Management (CRM)
 - Kundenbeziehungsmanagement
 - Supply-Chain-Management (SCM)
 - Lieferkettenmanagement

Standardsoftware

1. Softwareauswahl

1. Projektvorbereitung
2. Analyse
3. Anforderungen
4. Vorauswahl
5. Endauswahl

2. Einführung von Standardsoftware

3. Software-Lizenzmanagement

1. Lizenzformen
2. Lizenzmodelle
3. Beispiel: Microsoft
4. Beispiel: indirekte Benutzer
5. Beispiele im Outsourcing und Cloud-Computing
6. Folgen fehlerhafter Lizenzierungen
7. Ziele und Aufgaben des Lizenzmanagements

Softwareauswahl

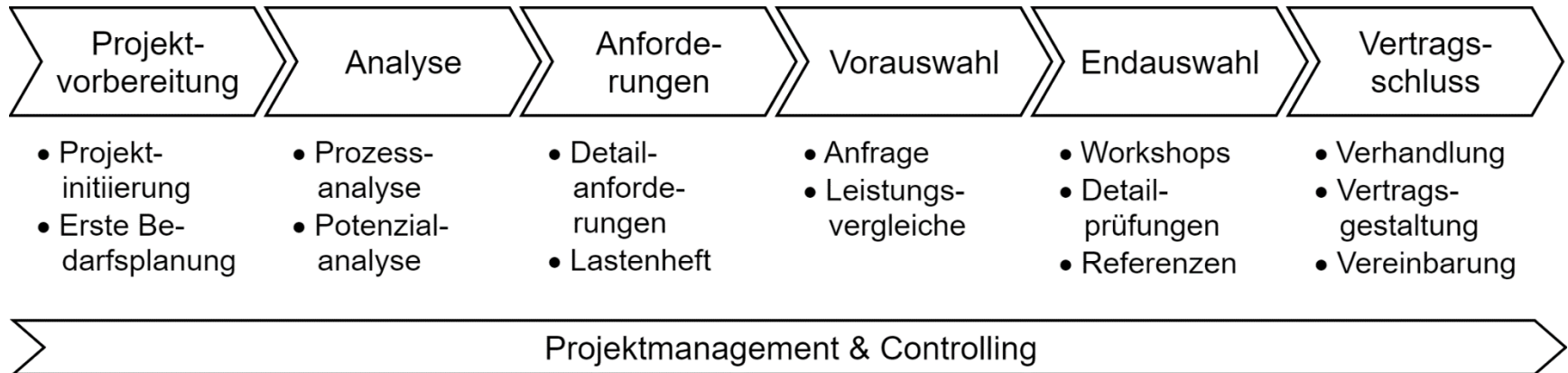
■ Sorgfältiger Auswahlprozess erforderlich

- Hohe Kosten
- Standardsoftware spielt eine wichtige Rolle für das Unternehmen und seinen Erfolg.
 - Geschäftsprozesse effektiv abwickeln
 - Planungen durch geeignete Verfahren unterstützen
 - Daten zur Entscheidungsunterstützung bereitstellen
- Ungeeignete Software führt zu ...
 - hohem Anpassungsbedarf,
 - unzufriedenen Benutzern,
 - ineffizienten Prozessen usw.

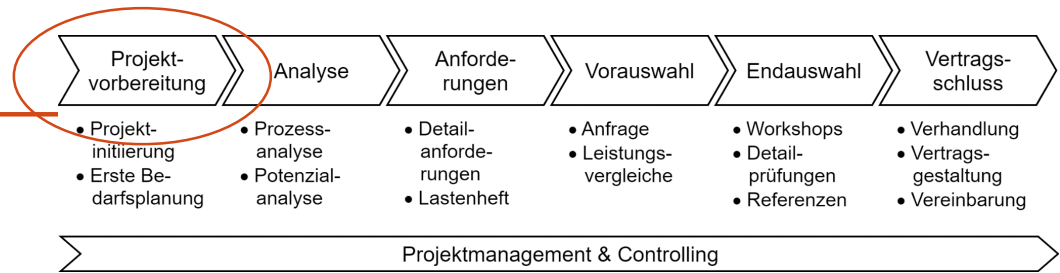
■ Größere Softwarebeschaffung

- Als Projekt organisiert und gesteuert
- Externe Berater mit Marktkenntnissen und Auswahlerfahrung einbeziehen

Phasenmodell der Softwareauswahl



Projektvorbereitung



■ Auswahlteam

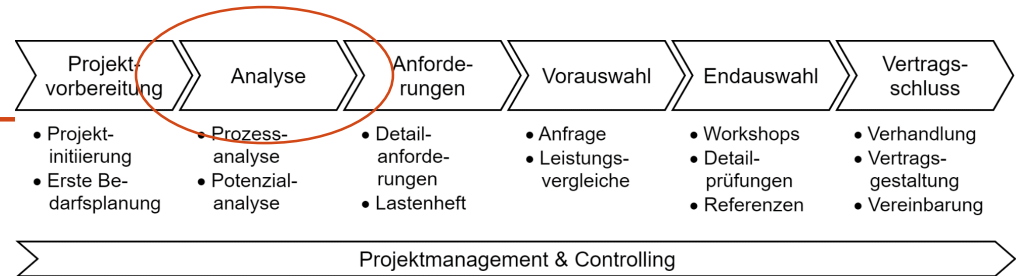
zusammenstellen

- Alle relevanten Stakeholder: Fachbereiche, IT, Management

■ Ziele festlegen

- Gemeinsam festlegen und schriftlich dokumentieren
 - Sonst geraten die Ziele aus den Augen und Beteiligte verfolgen ihre eigenen Interessen.
- Grundlage: IT-Strategie und EAM-Dokumentationen
 - Beispiel: Anvisierten Funktionsumfang mit Hilfe einer Capability-Map abstecken
 - Falls Dokumente und Pläne nicht mehr aktuell: Strategie überarbeiten und Dokumentationen aktualisieren

Analyse



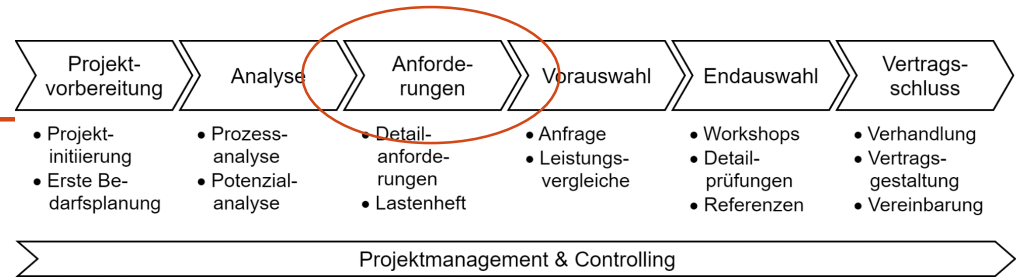
■ Geschäftsprozesse analysieren

- Grundlage: Prozesslandkarte und existierende Beschreibungen
- Für das Projekt relevante Prozesse genauer erheben, z. B. in einem Workshop
- Ziel: Prozesse verstehen und Anforderungen ableiten
 - Keine zu detaillierte Modellierung der Ist-Prozesse
 - Auch keine zu starre Festlegung der Soll-Prozesse, da sie später an die Standardprozesse der Software angepasst werden müssen
 - Standardsoftware enthält „Best Practices“
 - Insbesondere bei rein unterstützenden Prozessen ist es meist sinnvoll, nicht von den Standardprozessen abzuweichen

■ Wirtschaftlichkeit ermitteln

- Berechnung des Return-on-Investment (ROI)
- Finanzielle Vorteile entstehen z. B. durch Umsatzsteigerungen oder Einsparungen
- Klar festlegen, welche Vorteile erwartet werden
 - Während des Auswahlprozesses überprüfen, ob diese Vorteile tatsächlich realisierbar sind
 - Beispiel: Braucht man mit dem System für einen bestimmten Prozess tatsächlich weniger Arbeitszeit?
- Kosten
 - Lizenzgebühren, Hardware, Wartung, Schulungen, Personalkosten für Einführung, Anpassung und Erweiterung

Anforderungen



■ Wichtige Arten von Anforderungen

- Funktionale Anforderungen – aus Geschäftsprozessen ermitteln
- Anforderungen an den Anbieter
- Umsetzungsmethodik und eingesetzte Werkzeuge
- Technische Anforderungen
 - Z. B. Hardware, unterstützte Betriebssysteme, Datenbanken, Browserversionen.
- Schnittstellen- und Integrationsanforderungen

Kriterienkataloge

■ Vorgefertigte Kataloge

- Von Beratern oder auf Internetplattformen zur Softwareauswahl
- An eigene Bedürfnisse anpassen

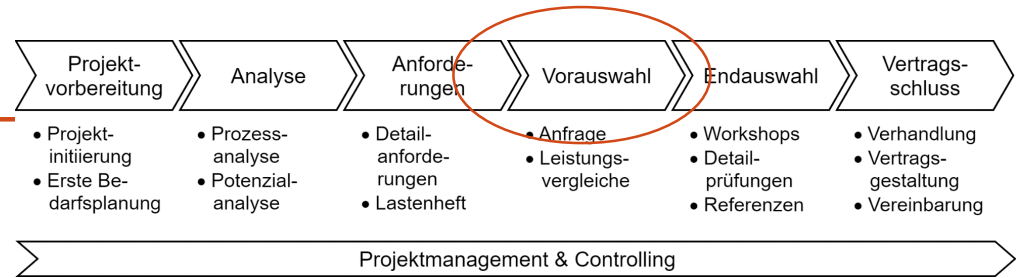
■ Kriterien

- Nicht möglichst viele Kriterien, sondern die für das eigene Unternehmen wichtigen
 - Sonst gewinnt das System mit dem größten Funktionsumfang, nicht das am besten passende.
- Kriterien gewichten
- K.O.-Kriterien: Müssen unbedingt erfüllt sein
- Geschlossene Fragen mit festgelegten Antwortmöglichkeiten (wegen Vergleichbarkeit), z. B.:
 - „Im Standard enthalten“
 - „Kleine Anpassung erforderlich“
 - „Große Anpassung erforderlich (mehrere Tage Aufwand)“
 - „Durch Partnerprodukt gelöst“
 - „Umsetzung nicht möglich“

Ausschnitt aus einem Kriterienkatalog für ERP-Systeme

- Allgemeines zum Projekt
 - Ausrichtung
 - Systemtechnik
 - Anlagenbuchhaltung
 - Finanzbuchhaltung
 - Kostenrechnung
 - Auftragsbearbeitung
 - Vertreterverwaltung (8 Fragen)
 - Kundenverwaltung (7 Fragen)
 - *Neuanlage eines Kunden bei Auftragserfassung*
 - *Einmalkunden*
 - *Auftragssperre*
 - *Kreditlimit*
 - *Sammelrechnungen*
 - *Verkaufskontrakte*
 - *ABC-Klassifizierung*
 - Lieferantenverwaltung (4 Fragen)
 - Artikelverwaltung (10 Fragen)
 - Preisfindung (12 Fragen)
 - Angebotserfassung (5 Fragen)
 - Auftragserfassung (11 Fragen)
 - Bestellwesen (9 Fragen)
 - Fakturierung (8 Fragen)
 - Inventur (6 Fragen)
 - Auslieferung (6 Fragen)
 - Lagerverwaltung (11 Fragen)
 - Materialwirtschaft
 - Produktionsplanung und -steuerung
 - Prozessfertigung
 - Personalabrechnung
 - IT-Outsourcing
- Insgesamt
über 500
Einzelfragen

Vorauswahl



■ Aussichtsreiche Anbieter auswählen

- Internetrecherche, Anbieter-Webinare, Marktstudien, Messebesuche, neutrale Berater

■ Reduktion der Anbieterzahl (Trichtermodell)

- Ausschluss durch K.O.-Kriterien, anschließend die interessantesten identifizieren

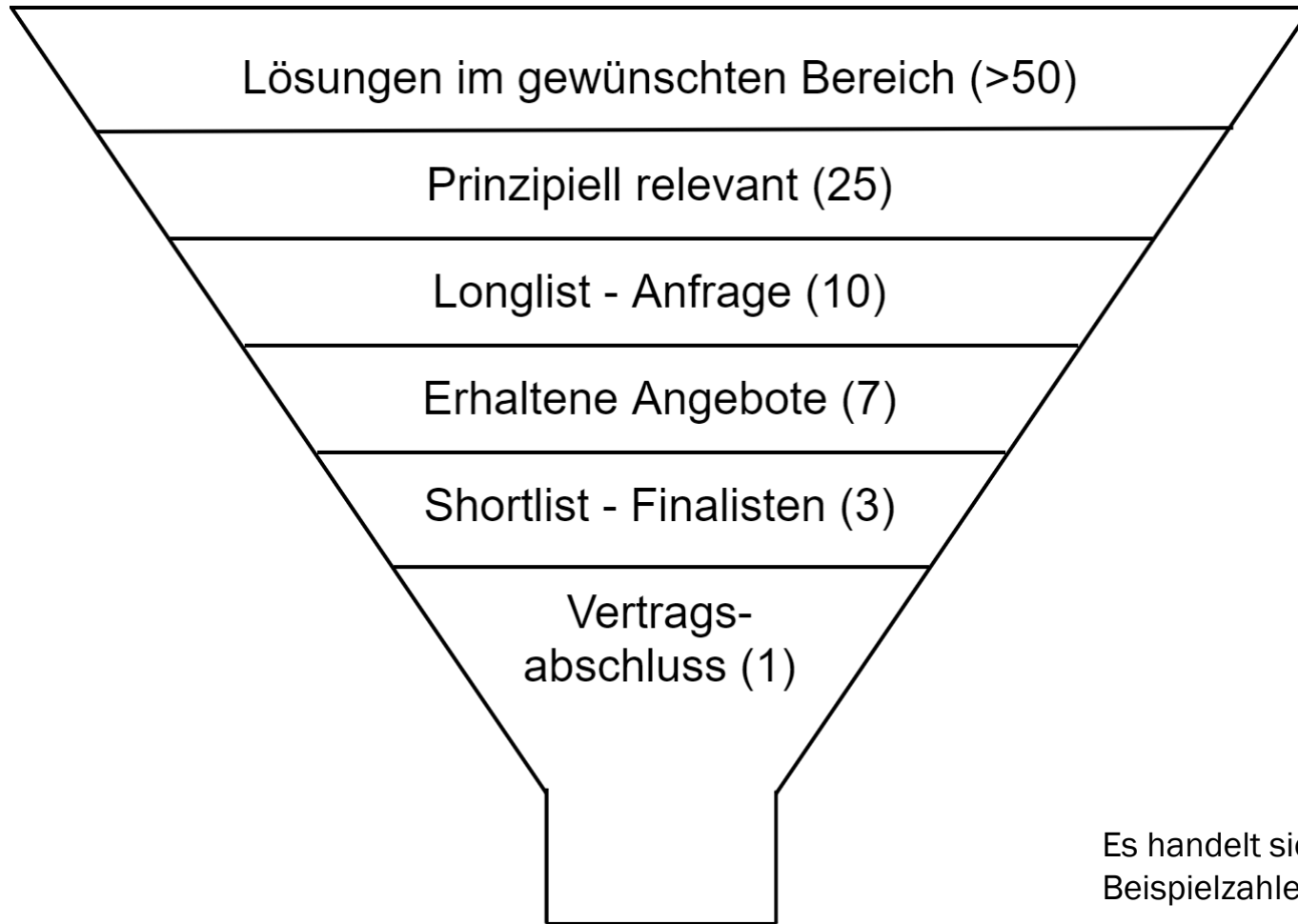
■ Longlist

- Diese Anbieter erhalten eine Anfrage mit Projektbeschreibung und Fragekatalog.
- Nicht zu viele Anbieter, sonst wird der Aufwand zu groß

■ Shortlist

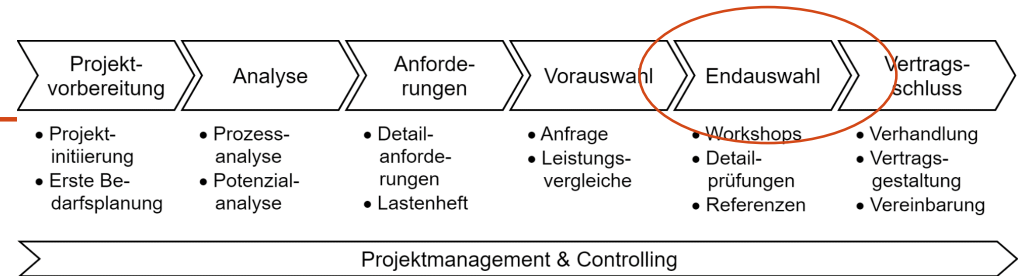
- Auswahl anhand ausgefüllter Fragebögen (mittels Kriteriengewichtung)
- Einladung zu Workshop

Trichtermodell zur Auswahl eines Softwareanbieters



Es handelt sich um
Beispielzahlen.

Endauswahl (1)



■ Workshops

- Nicht mehr als zwei bis vier Anbieter
- Alle relevanten Stakeholder des Kunden sollten teilnehmen.

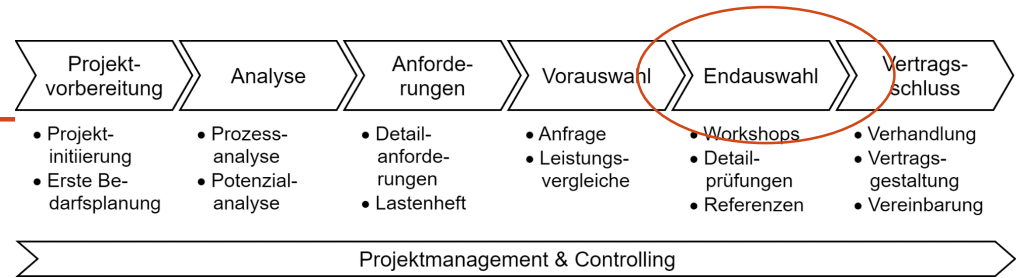
■ Gute Vorbereitung wichtig

- Sonst halten die Anbieter ihre Standardpräsentationen
- Den Anbietern einen Beispielprozess zur Verfügung stellen, mit der Bitte, ihn im Workshop mit ihrer Software zu zeigen
 - An dem Beispielprozess sollten sich wichtige Anforderungen demonstrieren lassen
- Videokonferenz im Vorfeld
 - Sicherstellen, dass der Anbieter das Szenario und die relevanten Inhalte vorbereitet hat

■ Testzugang

- Bei Cloud-Anwendungen
- Eigenes Ausprobieren der Software

Endauswahl (2)



■ Bewertung der Anbieter-Präsentationen

- Durch die Workshop-Teilnehmer nach vorher festgelegten Kriterien

■ Referenzkundenbesuche

- Bei anderen Kunden des favorisierten Systems
- Erfahrungen mit dem System und dem Anbieter erfragen

■ Prozessworkshop

- Vor dem endgültigen Vertragsschluss
- Abbildung der gewünschten Prozesse durchsprechen
- Erforderliche Anpassungen und Modifikationen identifizieren
- Terminplan für die Einführung aufstellen
- Grundlage für genaueres, endgültiges Angebot
- Ergebnisse in einem Lastenheft festhalten (wird Vertragsbestandteil)

Standardsoftware

1. Softwareauswahl

1. Projektvorbereitung
2. Analyse
3. Anforderungen
4. Vorauswahl
5. Endauswahl

2. Einführung von Standardsoftware

3. Software-Lizenzmanagement

1. Lizenzformen
2. Lizenzmodelle
3. Beispiel: Microsoft
4. Beispiel: indirekte Benutzer
5. Beispiele im Outsourcing und Cloud-Computing
6. Folgen fehlerhafter Lizenzierungen
7. Ziele und Aufgaben des Lizenzmanagements

Einführung von Standardsoftware

■ Einführungsprojekt

- Führt das Auswahlprojekt fort
- Prozesse umsetzen, Einstellungen vornehmen, Daten übertragen
- Externe Berater einbeziehen, die das System gut kennen

■ Projektmanagement-Aufgaben

- Projektplanung, Projektorganisation, Projektsteuerung und -überwachung, Qualitätsmanagement, Projektkommunikation, ...
- Veränderungsmanagement
 - Änderungen hinsichtlich Aufgaben, Verantwortlichkeiten, organisatorischen Zuordnungen, ...

■ Hersteller bieten Hilfsmittel an

- Vorgehensmodelle, Prozess-Referenzmodelle, Leitfäden, Softwaretools für Parameter-Einstellungen
- Vorgehensmodell muss meist angepasst werden
 - Auch Standardsoftwareprojekte werden zunehmend agil durchgeführt.

Feinkonzept

- **Legt die genaue Umsetzung der Prozesse und Anforderungen im System fest.**
- **Hierzu gehören**
 - Auswahl und Anpassung der benötigten Funktionalität
 - Globale Einstellungen (Sprachen, Währungen etc.)
 - Anpassung der Oberflächen
 - Organisationsstrukturen und Benutzerberechtigungen
 - Formulare und Berichte
 - Erweiterung von Datenstrukturen
 - Konfiguration und ggf. Programmierung von Schnittstellen
 - Programmierung von Erweiterungen
 - Vorgehen und Regeln zur Übernahme von Daten aus Altsystemen

Customizing (1)

- **Customizing: Anpassung an das jeweilige Unternehmen**
 - Durch Einstellung zahlreicher Parameter

- **Beispiel für einen Customizing-Parameter**
 - Wertermittlung von Materialien in einem bestimmten ERP-System
 - Bestimmung des aktuellen Wertes von Lagerbeständen
 - Hierzu benötigt man den Preis der Ware
 - Was, wenn verschiedene Teilmengen zu verschiedenen Preisen eingekauft wurden?
 - Im System vorhandene Methoden der Preissteuerung:
 1. Fester Standardpreis
 2. Ermittlung eines Durchschnittspreises aus verschiedenen Einkäufen
 - Im Customizing wird für jede Materialart (Fertigprodukt, Rohstoff, ...) eingetragen:
 1. Welches der beiden Preissteuerungsverfahren verwendet wird
 2. Ob das Verfahren für einzelne Materialien geändert werden darf

Customizing (2)

- **Was, wenn der Standard nicht ausreicht?**
 - Beispiel: Der Materialpreis soll nach einem ganz anderen Verfahren ermittelt werden.
 - Kann nicht durch Customizing gelöst werden
 - Möglichkeiten:
 - Preis wird außerhalb des Systems ermittelt und manuell eingetragen (Systembruch!).
 - Zusatzprogrammierung
 - Aufwändig
 - Bei späteren Releasewechseln muss die Erweiterung erneut getestet werden.

- **Customizing-Parameter**
 - Je nach System gibt es einige Hundert bis zu mehreren Tausend Parametern.
 - Korrekte Einstellung ist wichtig, damit das System richtig funktioniert.

Vor der Produktivsetzung („Go-Live“)

- **Testen**
 - U. a. Funktionalität, Berechtigungen, Formulare und Berichte, Schnittstellen, Lasttests
- **Datenübernahme aus Altsystemen**
 - Häufig haben Altsysteme keine Standardschnittstellen.
 - Andere Datenstrukturen, oftmals schlechte Datenqualität
 - Ggf. muss eigens eine neue Schnittstelle programmiert werden.
 - Mapping-Tabellen für die Umwandlung in die neue Datenstruktur
 - Verbesserung der Datenqualität mit geeigneten Tools
- **Schulung der künftigen Benutzer**
- **Key-User**
 - Ausgewählte Benutzer frühzeitig in das Projekt involvieren
 - Testen des Prototypen
 - Erste Ansprechpartner für Fragen der Kollegen

Strategien zur Umstellung auf das neue Produktivsystem

■ Stufenweise

- Benutzer können sich nach und nach mit dem System vertraut machen.
- Vor der Freischaltung weiterer Stufen können auftretende Probleme gelöst werden.
- Geringeres Risiko, da das Altsystem noch zur Verfügung steht
- Verschiedene Möglichkeiten:
 - Neues System zuerst nur für bestimmte Geschäftsobjekte oder Funktionalitäten
 - Nachteil: Parallelarbeit mit zwei Systemen, unterschiedliche Datenbestände müssen wieder zusammengeführt werden.
 - Neues System erst in einem einzelnen Bereich (z. B. einer Niederlassung) einführen
 - Erst, wenn es dort gut funktioniert, in das Gesamtunternehmen ausrollen

■ Big Bang (komplett zu einem Zeitpunkt)

- Höheres Risiko wegen Abschaltung des Altsystems
- Dafür entfallen Probleme des Parallelbetriebs.
- Risiko durch ausführliche Test- und Pilotphase reduzieren

Standardsoftware

1. Softwareauswahl

1. Projektvorbereitung
2. Analyse
3. Anforderungen
4. Vorauswahl
5. Endauswahl

2. Einführung von Standardsoftware

3. Software-Lizenzmanagement

1. Lizenzformen
2. Lizenzmodelle
3. Beispiel: Microsoft
4. Beispiel: indirekte Benutzer
5. Beispiele im Outsourcing und Cloud-Computing
6. Folgen fehlerhafter Lizenzierungen
7. Ziele und Aufgaben des Lizenzmanagements

Software-Lizenzmanagement

- **Software ist ein immaterieller Vermögenswert**
 - Kauf oder Miete von Software:
Erwerb von Nutzungsrechten, geregelt in einem Lizenzvertrag
 - Es gibt eine Vielzahl, z. T. sehr komplexer Lizenzmodelle

- **Software-Lizenzmanagement**
 - Verwaltung der im Unternehmen genutzten Software-Lizenzen
 - Sicherstellen, dass Lizenzbedingungen eingehalten werden
 - Software-Einsatz optimieren
 - Bei Software-Beschaffung
 - Bedingungen und Preismodelle der Anbieter analysieren
 - Bei Architekturentscheidungen (z. B. Nutzung virtueller Maschinen)
 - Auswirkungen auf Lizenzkosten prüfen

Wichtigste Lizenzformen (typische Eigenschaften)

- **Proprietäre oder kommerzielle Software**
 - In der Regel kostenpflichtig.
 - Quelltext nicht offengelegt, Veränderungen nicht zulässig

- **Freeware**
 - Kostenlose Nutzung, darf kopiert und weitergegeben werden
 - Manche Software ist nur für privaten Gebrauch kostenlos.

- **Shareware**
 - Proprietäre Software, darf meist frei verbreitet werden
 - Für bestimmte Zwecke oder einen begrenzten Zeitraum kostenlos, ansonsten kostenpflichtig

- **Open-Source-Software**
 - Quellcode liegt offen, kostenlose Nutzung, Verbreitung.
 - Darf angepasst werden.
 - Unterliegt einer Lizenz, deren Bedingungen eingehalten werden müssen. Meist Standard-Lizenzbedingungen genutzt, z. B. GNU General Public License (GPL) der Free Software Foundation.
 - Kommerzieller Verkauf von abgeleiteter Software oftmals eingeschränkt (z. B. auch nur als Open Source zulässig)

Lizenzformen

■ Zahlreiche Varianten

- Z. B. Software mit kostenloser Open-Source- und kostenpflichtiger Variante (mit zusätzlichen Eigenschaften)

■ Cloud-Software

- Vergleichbare Varianten
- Z. B. kostenlose Testphase
- Freemium-Modell
 - Kostenlose Accounts mit eingeschränktem Funktionsumfang
 - Für weitere Funktionen muss man auf einen kostenpflichtigen Account wechseln

Übersicht über gebräuchliche Lizenzformen

	Kostenlos	Kopier- und weiterverbreitbar	Uneingeschränkt nutzbar	Quelltext offen & modifizierbar	Unterliegt einer Lizenz
Open-Source	✓	✓	✓	✓	✓
Proprietäre Software					✓
Freeware	✓	✓			✓
Shareware		✓			✓

Quelle: Groll, T.: Lizenzmanagement in IT-Umgebungen.
In: Tiemeyer, E. (Hrsg.): Handbuch IT-Management. Konzepte, Methoden, Lösungen und Arbeitshilfen für die Praxis.
7. Auflage. München 2020, S. 695-740.

Lizenzmodelle

■ Typische Faktoren u. a.

- Lizenzart (z. B. Einzellizenz, Mehrplatzlizenz, Unternehmenslizenz)
- Lizenzklasse (z. B. Vollversion, Upgrade-Version, Versionen mit unterschiedlich nutzbarem Funktionsumfang)
- Lizenzmetrik (z. B. Zahl der namentlich festgelegten Benutzer, Zahl der gleichzeitigen Benutzer, Zahl der Geräte, Zahl der monatlich gedruckten Seiten)
- Lizenzbindungen (z. B. feste Zuordnung zu einem bestimmten Gerät oder Ausschluss der Nutzung in Cloud-Umgebungen)
- Laufzeiten (zeitlich begrenzt, unbegrenzt)

Software kaufen oder mieten

■ Softwarekauf

- Früher der Standardfall für On-Premise-Software (im eigenen Unternehmen installiert)
- Einmalige Lizenzgebühr, zeitlich unbegrenztes Nutzungsrecht
- Wartungsvertrag für Hotline, Störungsbehebungen, kleinere Updates
 - Jährliche Zahlung, Größenordnung: 20% des Kaufpreises
- Für ältere Softwareversionen wird der Support irgendwann eingestellt.
 - Kostenpflichtige Updates erforderlich um weitere Fehlerhebung, Sicherheitsupdates etc. zu erhalten (auch wenn man eigentlich mit den Features der alten Version zufrieden ist)

■ Software mieten

- Monatliche Zahlungen
- In der Cloud der Standardfall, aber zunehmend auch für On-Premise-Software

■ Lizenzmodelle und –bedingungen sind oft sehr komplex

- Auswahl des geeigneten Lizenzmodells und Abschätzung der entstehenden Kosten schwierig

Beispiel: Microsoft Windows-Server

- Lizenzkosten sind abhängig von der Zahl der physischen Kerne (Cores) pro Serverprozessor (CPU) und der Gesamtzahl der Kerne in einem Server.
- Mindestlizenzmengen 8 Kerne pro CPU und 16 pro Server
- Kann in Stückelungen von je 2 oder 16 Core-Lizenzen gekauft werden.
- Zahl der ausführbaren Betriebssystemumgebungen ist beschränkt (physikalische und virtuelle Umgebungen werden unterschieden).
 - Ab einer gewissen Menge ist eine andere Edition vorteilhaft.
- Für Client-Programme, die auf den Server zugreifen, sind zusätzlich Client-Access-Lizenzen erforderlich.
 - Können für Endgeräte oder natürliche Personen erworben werden

Auswirkungen von Architekturentscheidungen

- Architekturentscheidungen können sich auf Lizenzkosten auswirken.
- Im Windows-Server-Beispiel:
 - Ersetzt man einen älteren durch einen neueren Server mit mehr Prozessorkernen, schnellen die Kosten für die Server-Software in die Höhe, obwohl sich an der Nutzung nichts ändert.
 - Einsatz virtueller Maschinen kann teuer werden, wenn sie dynamisch auf unterschiedliche physische Maschinen verteilt werden.
 - Lizenzen für die Kerne all dieser physischen Maschinen nötig

Beispiel: Indirekte Nutzer

- **Lizenzkosten sind häufig von der Nutzerzahl abhängig**
 - Z. T. unterschiedliche Arten von Nutzer-Accounts mit unterschiedlichen Funktionsumfängen

- **Wenn ein Drittsystem Daten mit der betreffenden Software austauscht**
 - Unter Umständen betrachtet der Hersteller alle Nutzer des Drittsystems auch als Nutzer seines Systems.
 - Handelt es sich beispielsweise um ein Portal für alle Mitarbeiter und Kunden, so explodieren die Lizenzkosten.
 - Faire Regelung für indirekte Benutzer nötig
 - Unter Umständen wird dies bei der Einführung der Software noch gar nicht betrachtet, da der Bedarf erst später entsteht.

Beispiele im Outsourcing und Cloud-Computing

- **Die vom Kunden erworbene Software läuft auf der Hardware eines Outsourcing-Providers.**
 - Zulässigkeit klären
 - Hardwareoptimierungen des Providers, von denen der Kunde gar nichts mitbekommt, können sich auf die benötigten Lizenzen auswirken.

- **Cloud-Computing**
 - Zeit- und nutzungsabhängige Preismodelle sind im Idealfall einfacher, z. B. feste Monatsgebühr pro User.
 - Z. T. gemischte Preismodelle oder kombinierte Verbrauchswerte (z. B. gespeicherte Datenmenge und monatliche Zahl der Zugriffe)
 - Oftmals werden verschiedenen einzelne Cloud-Services mit verschiedenen Preismodellen kombiniert.
 - Entstehende Kosten sind schwer abzuschätzen.
 - Hybride Szenarien, z. B. eigene Hardware kombiniert mit Cloud-Infrastruktur, können komplizierte Lizenzfragen aufwerfen.

Lizenzmanagement

- **Teil des IT-Asset-Managements**
 - Verwaltet sämtliche IT-bezogenen Vermögenswerte
- **Eng mit Service-Configuration-Management verknüpft**
 - Verwaltet Informationen zu technischen Details.
 - Liefert für das Lizenzmanagement z. B. die Information, wo welche Software installiert ist.

Wichtige Ziele des Lizenzmanagements

- **Transparenz schaffen**
- **Compliance herstellen**
- **Einsparpotenzial aufzeigen**
- **Strafzahlungen vermeiden**
- **Softwarebeschaffung unterstützen**
- **Herstelleraudits vorbereiten**

Daten für das Lizenzmanagement

- **Arten von Daten**
 - Technische Daten über installierte Software
 - Lassen sich z. T. automatisiert auslesen
 - Kaufmännische Daten
 - Abgeschlossene Lizenzverträge mit Bedingungen und Kosten

- **Vergleich der tatsächlich installierten Software mit den erworbenen Lizenzen**
 - Auswertung in Form eines Compliance-Reports
 - Führt Über- und Unterlizenzierungen auf
 - Hilfreich bei Hersteller audits
 - Hersteller hat keine Handhabe, anhand von Schätzungen vermeintliche Unterlizenzierungen zu reklamieren.

Administrative Aufgaben des Lizenzmanagements

- Festlegung und Überwachung von Regeln für die Beschaffung, Nutzung, Weiter- und Rückgabe von Softwarelizenzen
- Erfassung, Verwaltung und Überwachung der Lizenzverträge
- Erfassung des Lizenzbedarfs
- Erfassung von technischen Voraussetzungen und Abhängigkeiten der zugehörigen Software
- Zuordnung von Lizenzen zu Personen, Organisationseinheiten oder Hardware
- Wertermittlung und Wertfortschreibung der Lizenzen
- Verwertung und Weiterverkauf der Lizenzen
- Klärung juristischer Fragen

Aufgaben im Software-Lebenszyklus

- **Lizenzmanagement betrifft den gesamten Software-Lebenszyklus**
 - Anforderung, Beschaffung, Installation, Betrieb, Entsorgung

- **Beispiele**
 - Software-Auswahl
 - Prüfen, wie angebotene Lizenzmodelle zu den Anforderungen passen
 - Beschaffung
 - Lizenzverträge prüfen und archivieren
 - Nutzung
 - Weitergabe an andere Nutzer:
 - Auf Lizenzkonformität achten
 - Weitergabe dokumentieren
 - Kostenmanagement
 - Softwarelizenzkosten sind ein großer Kostenblock
 - Ihre Erfassung und Zuordnung sind daher Voraussetzungen für die IT-Kostenrechnung.