

Vorlesung 3

Digitaler Zwilling

- digitale Darstellung eines realen Objekts oder Systems (materiell oder immateriell)

Integrationsansätze

- **Datenintegration:** Datenbestände von mehreren Informationssystemen werden zentral gespeichert (nicht mehrfach)
- **Funktionsintegration:** mehrere Funktionen werden in einem Informationssystem gebündelt
- **Prozess- oder Vorgangsintegration:** Prozesse folgender Funktionalitäten sind nahtlos miteinander verbunden (Schnittstellen)

SAP

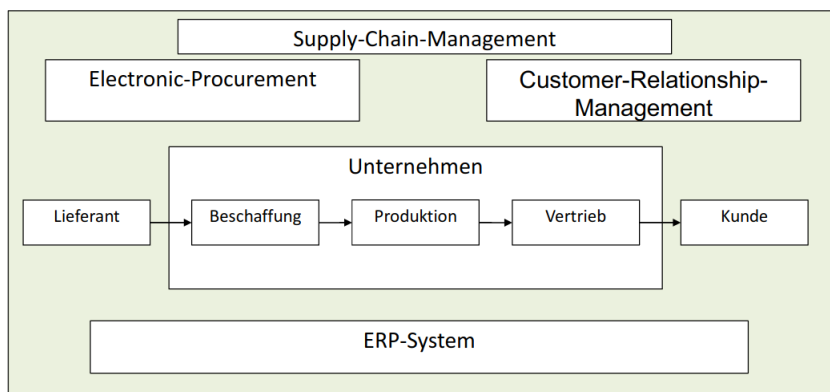
- Kundenauftrag wird erfasst
- automatisches Ausführen von: Bestellung der Rohmaterialien, Erzeugung von Fertigungsaufträgen, Übermittlung an die Finanzplanung
- Rollen & Rechte verteilen

ERP Systeme

→ integrierte betriebswirtschaftliche Softwarelösungen, die eine Vielzahl Geschäftsprozesse eines Unternehmens abdecken

- hohe Datenintegration: zentrale Datenbank
- hohe Funktions- und Prozessintegration: Schnittstellen

Informationssysteme in der Praxis: Enterprise Resource Planning (ERP)



Systembereitstellung – Goldene Regeln

	1. Software ist nie fertig.	<ul style="list-style-type: none"> ▪ Laufende Aktualisierungen ▪ iterative Verbesserungsprozesse ▪ Anpassung an Kundenbedürfnisse
	2. Software ist eine Teamleistung, niemand kann alles machen.	<ul style="list-style-type: none"> ▪ Komplexität erfordert Aufgabenteilung (Projekt) ▪ Heterogene Expertise (Domäne und Entwicklung) ▪ Verwendung von Vorgehensmodellen
	3. Großartiges entsteht durch Tausende kleiner Verbesserungen.	<ul style="list-style-type: none"> ▪ Inkrementelle Verbesserungen ▪ Modulare Architektur
	4. Software läuft nicht von selbst.	<ul style="list-style-type: none"> ▪ Ständige Überwachung ▪ Veränderung und Aktualisierung
	5. Komplexe Systeme benötigen DevOps, um gut zu laufen.	<ul style="list-style-type: none"> ▪ Kontinuierliche Veränderungen ▪ Kontinuierliches Testen ▪ Kontinuierliches Verbessern

Softwareindustrie

- Nutzen eines Programms steigt mit Anzahl der Nutzer
- keine Vervielfältigungskosten: Software kann mehrfach verkauft werden ohne Mehraufwand
- kein Wertverlust durch Gebrauch

Make or Buy

Eigenentwickelte Software („ Make “)	Fremdentwickelte Software („ Buy “)
<ul style="list-style-type: none"> • Nahezu vollständige Abdeckung unternehmensspezifischer Anforderungen • vollständige Integration in die Gesamtheit bereits implementierter Anwendungen • Kosten für Anpassung und Einführung entfallen weitestgehend 	<ul style="list-style-type: none"> • Eliminierung der Entwicklungszeiten durch rasche Produktverfügbarkeit • Reduzierung der Einführungs- und Übergangszeit im Vergleich zu Individual-Software • Gewährleistung der Weiterentwicklung durch den Anbieter • Unabhängigkeit der Programmentwicklung von der Verfügbarkeit der IT-Ressourcen

Nutzenkategorien von Informationssystemen

	monetär bewertbar	nicht monetär bewertbar
quantifizierbarer Nutzen	<ul style="list-style-type: none"> • Verkürzung von Bearbeitungszeiten • Abbau von Überstunden • Materialeinsparung • Personalreduzierung 	<ul style="list-style-type: none"> • Schnellere Angebotsbearbeitung • Weniger Terminüberschreitungen • Höherer Servicegrad • Weniger Kundenreklamationen
nicht quantifizierbarer Nutzen		<ul style="list-style-type: none"> • Erhöhung der Datenaktualität • Verbesserte Informationen • Gesteigertes Unternehmensimage • Erweiterte Märkte und Geschäftsfelder

Anwendungslebenszyklus

- Entwicklung
- Einführung
- Wachstum
- Sättigung/Reife
- Rückgang
- Abschaffung

Planung eines Softwareentwicklungsprozesses

- Anforderungsanalyse und Erstellung einer Spezifikation
- Design
- Entwicklung
- Test und Integration

- Auslieferung des Produkts
- Wartung und Support

Strukturgetriebene Softwareentwicklung: Spiralmodell

Wiederholender Durchlauf von Entwicklungsphasen in Iterationen von jeweils 4 Schritten mit kontinuierlicher Bereitstellung von Prototypen.

- (1) **Analyse:** Definition von Rahmenbedingungen, Zielen, Anforderungen und Lösungsalternativen, Freigabe zur Umsetzung
- (2) **Evaluierung:** Evaluierung der umgesetzten Lösungsalternativen. Darauf basierend Erkennung von Risiken und Erarbeitung adäquater Strategien zur Vermeidung der Risiken.
- (3) **Realisierung:** Definition und anschließende Realisierung des Vorgehens, basierend auf den identifizierten Risiken.
- (4) **Planung:** Review der vorangegangenen Schritte und Planung der nächsten Iteration

Prinzipien agiler Softwareentwicklung

- Transparenz und Geschwindigkeit der Entwicklung erhöhen
- Fehler minimieren
- Kommunikation und Interaktion!

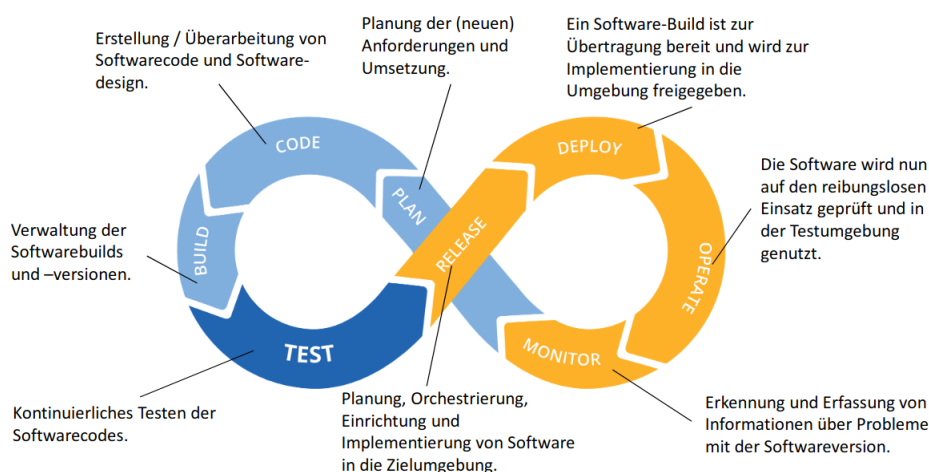
SCRUM

- Modell der agilen Softwareentwicklung
- Transparenz, Überprüfung und Anpassung
- grober, zeitlicher Rahmen wird definiert und dann angepasst
- Teams sind selbstorganisiert → Daily Meetings

DevOps

- Development + Operations
- Ziel: in sich verändernden Umgebungen mit schlanken und flexiblen Software-Entwicklungsprozessen schnell zu reagieren

DevOps zur Integration von Entwicklung und Betrieb



Limitationen und Herausforderungen von DevOps

- Flexibilität
- Automatisierung
- Lean-Prinzipien → System optimieren
- Alignment-Herausforderung → Überwachung der wichtigsten Indikatoren
- Kultur- und Wissensaustausch

„Magisches Dreieck“ des Projektmanagements

Erfolgsfaktor 1: Qualität & Funktionalität

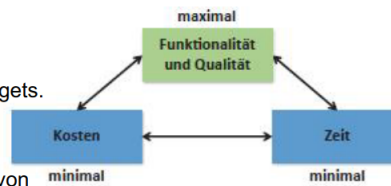
Präzise Definition der Funktionalität und Qualität sowie deren permanente Überprüfung.

Erfolgsfaktor 2: Kosten

Kostenkontrolle innerhalb des zugewiesenen Budgets.

Erfolgsfaktor 2: Zeit

Vermeidung von Verzögerungen und Festlegung von Projekt-Schritten („Milestones“)



Proprietäre vs. Open Source Software

Traditionelle Informationssysteme	Open Source Informationssysteme
Entwicklung durch Softwareunternehmen	Entwicklung durch Programmierer verschiedener Organisationen und Freiwillige
Quellcode verbleibt im Softwareunternehmen	Quellcode der Software ist öffentlich zugänglich
Verbesserungen und Fehlerbehebungen langwieriger	Zügige Verbesserungen und Fehlerbehebungen möglich
Lizenzmodelle: Organisationen erwerben Lizenz zum Betrieb der Software	Keine Lizenzkosten
Spezifische Kundenanpassung nicht möglich	Möglichkeit der Kundenanpassung gegeben

Vorgehen zur Softwareauswahl

- Ist-Analyse
- Definition der Anforderung
- Marktanalyse
- Vergleich der Angebote
- Vertragsverhandlung

IT-Outsourcing: Vor- und Nachteile

Vorteile	Nachteile
<ul style="list-style-type: none"> • Kosteneinsparungen • Flexibilität • Schnellere Umsetzungen • Keine Investitionsausgaben • Freies Personal zur Fokussierung auf Kernkompetenzen 	<ul style="list-style-type: none"> • Verlust der Management-Kontrolle • Abhängigkeit vom Dienstleister • Sicherheitsrisiken und geringe Kontrolle über Datengut • Hoher Kommunikationsaufwand • Verlust des Know-Hows