

Seibt: Wirtschaftsinformatik, #99

12.10.2005

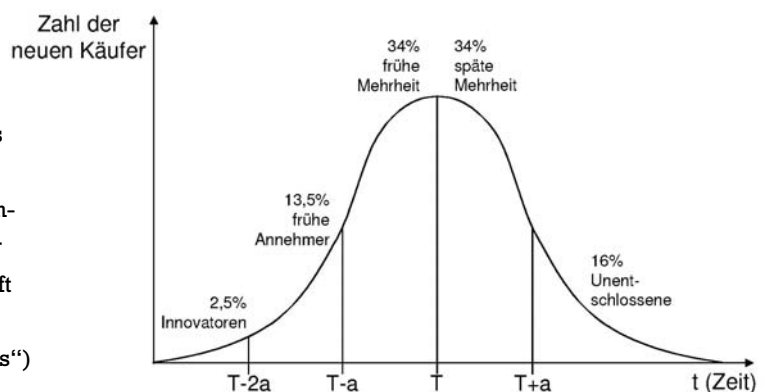
Die Vorlesung folgt dem Script Dr. Seibt's; die Infos hier dienen nur der Ergänzung.
Seitenzahlen entsprechen dem gedruckten Script, nicht den aktuell gezeigten PP-Seiten.

[138](#)

- „Querschnittssysteme“ werden in dieser Vorl. nicht behandelt, trotzdem sind sie wichtig; hier intensiver behandelt: Admin.- und Führungssysteme
- Controlling ist ein Problem: es ist unbeliebt bei den Kontrollierten; der Controller ist meist der oberste Finanz- und Rechnungswesen-Kontrolleur

[140](#)

- Lebenszyklus-Management:
 - Produktlebenszyklen, z.B. KFZ: bei einem neuen PKW nimmt das Verkaufsvolumen zu und später wieder ab – die Mischung persönlicher Erfahrung und technischer Unterstützung durch Software hilft Vorausberechnungen der „Verkaufskurve“ („Produktzyklus“)



- electronic commerce
 - INet-basierter Vertrieb
 - EDI (electronic data exchange)
 - Anbieter und Käufer bedienen sich des INets (= „elektronische Märkte“)
- Supply-Chain-Management
 - Hersteller – Distributor – Logistiker – Einzelhandel – Endkunden
 - JIT = Just In Time
 - Autobahn wird zum größten Lager

[141](#)

- Unter jeder BIS Organisation-/Menschen-Hülle gibt es mindestens 1 BAS
- Lebensdauer von BAS/BIS-Systemen ist immer begrenzt, – technisch und organisatorisch bedingt (oftmals ca. 4-5 J.); aber:
 - „don't change a running system“
 - evtl. Schnittstellen nötig für alte und neue Systeme
 - „Middleware“ (prüfungsrelevantes Thema)
 - Ende der Lebenszeit kündigt sich z.B. durch stark steigenden Aufwand für Wartung und Pflege an

6. IDV (individuelle Datenverarbeitung) vs. dezentrale Systementwicklung

- Zur Erstellung des PIV reicht schon die Konfiguration eines PC-Systems – das PIV muß nicht zwingend programmiert werden
→ Ggs.: „Customizing“ (=für alle Anwender des Betriebs) [144](#)
- Das PIV kann als individuelle BAS/BIS-Variante betrachtet werden
→ Unterschied zum „echten“ BAS/BIS ist die Unverbindlichkeit, da die Prozesse nur für den Ersteller gelten und größere Spielräume möglich sind
→ generell viel Spielraum für ein PIV bei Planungs- und Controlling-Systemen, kein Spielraum bei produktionsunterstützenden Systemen
- „Benutzer-Service“ zur Unterstützung der PIV's war getrennt von der zentralen DV angesiedelt [146](#)
- Keine IDV bei z.B. Personaldatenverwaltung, da es eine Vielzahl von Einschränkungen/ gesetzlichen Vorgaben gibt [147](#)
- Volumen der IDV muß Waage halten – übermäßige Nutzung der individuellen Möglichkeiten kann zu Wildwuchs und der Verschwendung von Arbeitszeit führen [150](#)
- Im Aufkeimen der IDV (vor 10-15 Jahren) gab es bei einige großen Fa. Belohnungen für die „beste IDV“ (RWE, Daimler, Siemens) [155](#)
- ODV (organisatorisch verbindliche DV) muß zumindest (organisatorisch) abteilungsweit verbindlich sein, damit sie Sinn macht
→ ODV als Ggs. zur IDV [156](#)
- Damit die Fachabteilungen wirklich autark werden, wird auch eigenes Know-How entwickelt – u.U. indem mit MA in denselben Weiterbildungskursen, in denen die Kollegen aus der IT-Abteilung sitzen [157](#)
- Mit steigenden Budgets der Fachabteilungen steigen die Möglichkeiten, von der IT-Abteilung unabhängig zu werden; zentrale IT-FA werden mehr und mehr ausgelagert und/oder geschlossen
→ IDS (weltgrößter Fremddienstleister dieser Art) [158](#)
- I&K-Technik = Informations- und Kommunikationstechnik [159](#)
- Allein die Tatsache, daß „der Kollege Müller“ die IDV realisiert hat, würde bei einer abteilungsweiten 1:1-Einführung zum Problem [162](#)

- ODV-Entwicklung in zentraler Anwendungsentwicklung dann, wenn Fachabteilung nicht genügend Budget hat

7. SAS (Standard-Anwendungssoftware)

- Gründe für IS:
 - Zeitprobleme bei Eigenentwicklung
 - spezifische Anforderungen

- Gründe für SAS:
 - hohe Verfügbarkeit
 - potentielle Vergünstigungen durch den Standard
 - werden u.U. durch notwendiges Customizing aufgeessen

- „Schränkware“ = Gekaufte Software, deren Funktionalität sich als unpassend erwies und die unbenutzt bleibt

- Deutsche Unternehmen legen Wert darauf, daß individuelle Betriebsspezifika in der Software abbildbar sind, amerikanische U. akzeptieren ein größeres Maß an Standardisierung

- Prozeß der Entwicklung eines BAS in 5 Stufen:
 1. Anforderungen ans zukünftige System festlegen („Requirements“) nach einem Business-Modell (z.B. ARIS, vgl. Script v. 5.10.)
 2. Analyse (der Anforderung) und Design (der Lösung) mit Definition DB, Screen, Aufgabenabwicklung („Functions“), Gesamtdesign („Architektur“)
 3. Programm(e) herstellen
 4. Module anhand ...
 - Logik (der Funktionen)
 - Screens
 - DB I/O
 - etc.... testen und gesamtintegrieren
 5. Übernahme in Realumgebung und mit echten Daten Produktion und Nutzung
...
Pflege, Wartung, Weiterentwicklung
 - führt dann evtl. wieder zu Pkt. 1.