

Agenda: Systembeschaffung

- **Wirtschaftlichkeitsbetrachtung bei Systembeschaffung**
- **Kosten- / Nutzenbetrachtung**
- **Wirtschaftlichkeitsbetrachtung**
- **Systemvergleich**
 - **Grundorientierung**
 - **Softwareauswahl / Softwareinspektion**
 - **Spezifizierung neuer Anforderungen**
 - **Maßnahmen zur Absicherung der Entscheidung**

Wirtschaftlichkeitbetrachtung bei Systembeschaffungen

- **Kosten/Nutzen nach Systementscheid**
 - Mitarbeiter haben häufig keine klaren Vorstellungen und vielfach auch keine gesicherte Datenbasis (!) über Art und Umfang anfallender Kosten
 - Erwartete oder erhoffte Nutzen sind wegen vielfach ungesicherter Datenbasis unscharf
 - Bei der Erstellung einer Kosten-/Nutzen-Analyse sind die Kosten/Nutzen oft stark beeinflusst von unterschiedlichen Interessenslagen
 - Mitarbeiter
 - Betrieb (Management)
 - eingeschaltete Berater

Wirtschaftlichkeitbetrachtung bei Systembeschaffungen

- Kosten/Nutzen nach Systementscheid
 - Frühzeitige Überlegungen zur Notwendigkeit und Sinnhaftigkeit einer Systembeschaffung
 - Entscheidung durch erfahrene EDV-Manager und Fachleute
 - Neutrale Kosten-/Nutzen-Darstellungen
 - Unterscheidung Einmalkosten, laufende Kosten
 - Einmalkosten differenzieren in Reihenfolge ihres Eintretens

Kosten-Kriterien (1)

- Vorbereitungs-/ Planungsphase
 - Planungskosten (interne und externe)
 - Projektorganisation (interne und externe)
 - Entwicklungskosten (falls Eigenentwicklung)
 - Anpassungskosten
 - Einführungskosten

Kostenelemente in früher Phase eines EDV-Projektes ohne Existenz eines ausreichenden Lasten- / Pflichtenheftes nur sehr grob und unzureichend abschätzbar

Kosten-Kriterien (2)

- Systemkosten (Hardware und Software)
 - neu zu beschaffende Hardware
 - vorhandene und zu integrierende Hardware
 - neu zu beschaffende Software
 - vorhandene und zu integrierende Software
 - Installationskosten (interne und externe, Datenleitungen nicht vergessen)
 - Materialkosten (bis zur Abnahme des Systems)

Kosten-Kriterien (3)

- Einführungskosten
 - Kosten für Systemtest (interne und externe, die beide häufig vergessen werden!)
 - Softwareanpassungen (bei der Integration "alter" mit "neuer" Software)
 - Übernahme von Altdatenbeständen (Migration)
 - Schulungskosten (externe, aber interne nicht vernachlässigen!)
 - Einarbeitungskosten (u.U. mehrfach anfallend, weil Plantermine nicht gehalten werden)
 - Einsparungspotential
 - Unvorhergesehenes

Kosten-Kriterien (4)

Laufende Kosten

- klare Kostenbestandteile
 - Raum- und Energiekosten
 - Personalkosten
 - Wartungskosten für Hardware und Standardsoftware
 - Versicherungen
- weniger klare Kostenbestandteile
 - Leitungskosten und Postgebühren
 - Update-Kosten
 - Fortbildungsmaßnahmen
 - Datenschutz und Datensicherung
 - Beratungskosten
 - Sachkosten, Verbrauchsmaterial usw.

Nutzen-Kriterien

- Häufig genannte Nutzen-Kriterien:
 - Produktivitätssteigerung
 - Zeitersparnis
 - Kosteneinsparungen
 - Personaleinsparungen
- Schwierigkeit: Nutzen sachgerecht und inhaltlich zu definieren und zu quantifizieren!
- Personaleinsparungen sind ein besonders sensibler Bereich (Mitbestimmungsregelungen)
- Betrachtung des häufigen Risikos der Projektverzögerung unter dem Aspekt des Gesamtrisikos

Quantitative Nutzen (1)

- Monetär nicht quantifizierbare Nutzen
 - Einschätzung mit Experten, um überhaupt eine (Vor-) Entscheidung treffen zu können
 - Unterscheidung zwischen unbedingten Notwendigkeit (z.B. Erfüllung gesetzlicher Vorgaben, Auflagen) und Notwendigkeiten aus dem Markt (Marktdruck, Markttrend, time to market)
- In den „Glaubensbereich“ gehören, solange nicht definiert und messbar, folgende Punkte:
 - Beschleunigung von Arbeitsabläufen
 - Leistungssteigerung bei der Aufgabenerstellung
 - bessere interne und externe Kommunikation
 - Erhöhung der Stabilität und Verfügbarkeit
 - Erhöhung der Flexibilität

Quantitative Nutzen (2)

- Weitere Punkte des „Glaubensbereichs“, die nur schwer definieren und messbar sind:
 - Qualitätsverbesserung
 - ISO 9000 zertifizierte Systemanbieter sollte nicht blindlings vertraut werden
 - Investitionen in die Zukunft
 - Hoffnungen auf Verbesserung der Auftragslage
 - Aufrechterhaltung wertvoller (Entwickler-) Kapazitäten
 - Einsicht der Investition in eine längst fällige technische Notwendigkeit

Oft wird in diesem Zusammenhang auch vom Paradigmenwechsel gesprochen

Quantitative Nutzen (3)

- Weitere Punkte des „Glaubensbereichs“, die nur schwer definier und messbar sind:
 - Kürzere ROI (Return of Investment) bei Investitionen
 - Verfall der Hardwarepreise, ständig verringerte System-Nutzungszeiten (Lifecycle)
 - Verlust des Anschluss an technische Entwicklung (und damit die Wettbewerbsfähigkeit in globalen Märkten!)
 - Outsourcing zur Kostenreduzierung
 - Konzentration auf Kernkompetenz
 - Allheilmittel Leasing

Wirtschaftlichkeitsbetrachtung (1)

- Bewertungsaspekte von Softwarelösungen:
 - Komplexität der zu bewältigenden Aufgabe
 - Voraussichtlicher / tatsächlicher Aufwand bei der Erstellung der Programme
 - Einbettung in die beim AG/AN vorhandenen Entwicklungsumgebung
 - Zeitraum der Entwicklung (allgemein verfügbare und erkennbare Technologiebasis)
 - Zukunftssicherheit (Portabilität und Marktentwicklung)
 - Zuverlässigkeit (Konsistenz in den Konzepten und Stabilität in der Anwendung)
 - Existenz von klar definierten Schnittstellen (zu anderen Programmen, DB usw.)

Wirtschaftlichkeitsbetrachtung (2)

- Ferner sind für einen Erfolg im Markt von Bedeutung:
 - Zielgruppe (Individual- oder Standardsoftware)
 - Zu erwartender Nutzen für den / die Anwender bzw. Lizenznehmer bei Weiterverkauf
 - Relation der Entwicklungskosten zu möglichen Preisen (im Wettbewerb)
 - Preis- / Leistungsverhältnis
 - Zeitpunkt des Markteintrittes

Wirtschaftlichkeitsbetrachtung (3)

- SW-Entwicklungen scheitern an unzureichenden und sich ändernden Vorgaben (Moved Targets)
- Fehleinschätzung, dass die bei Überschreitung von Terminen und Kosten entstehenden Mehraufwendungen später durch Weiterverkauf an Dritte kompensiert und sogar Gewinne erzielt werden
- Es existieren oft nur verschwommene Vorstellungen über erhoffte Zielmärkte, wobei die Aspekte "time to market" und Lebenszyklen von "Standard"-Softwareentwicklungen“ vernachlässigt werden

Systemvergleich (1)

- Allgemeines-1
 - Zur ersten Orientierung der Leistungsfähigkeit mehrerer Softwaresysteme werden üblicherweise Checklisten benutzt
 - Checklisten werden nach Funktionalitäten und Sachgebieten gegliedert
 - Je mehr Sachgebiete, um so breiter ist das Anwendungsspektrum
 - Je umfangreicher die Analyse getrieben wird, umso wahrscheinlicher ist die Gewinnung eines einigermaßen zutreffendes Bild über die voraussichtliche Eignung

Systemvergleich (2)

- Allgemeines-2
 - Checklisten spiegeln genaue Anforderungen eines konkreten Produktes und die "Highlights" der speziellen Anwenderlösung wieder
 - Vergleichbarkeit mehrerer Lösungen wird erschwert, wenn Analyse zu tief getrieben wird; eine zu wenig differenzierte Analyse gibt ebenfalls ein ungenaues Bild
 - Markierung einer Funktionalität (durch X = vorhanden) bedeutet nicht, dass sich die verglichenen Eigenschaften tatsächlich vollständig entsprechen. Es signalisiert für einen Sachverhalt eine vermutete Entsprechung
 - Verbesserung durch weitergehende Differenzierungen (X ++, X --) eröffnet dem Beurteiler einen zusätzlichen subjektiven Spielraum

Systemvergleich (3)

- Risiken bei der Systemauswahl
 - Systemvergleich mit Checklisten ist nicht immer objektiv wie es die äußere Form vermuten lässt.
 - Manipulationen durch (verdeckte) Grundorientierung an der Lösung und entsprechende Differenzierung der Funktionalität sind nicht nur möglich, sondern häufig sogar beabsichtigt
 - Checklistenvergleiche nach Möglichkeit nur als Orientierungshilfe nutzen

Systemvergleich (4)

- Vorgehensweise beim Systemvergleich
Es empfiehlt sich, folgende drei Phasen für einen Systemvergleich vorzusehen
 - Die Grundorientierung
 - Die Softwareinspektion
 - Die Spezifizierung neuer Anforderungen

Grundorientierung

- Auf Basis genauer Kenntnis einer Anwenderlösung wird eine Checkliste in ausreichender Darstellungsform und Darstellungstiefe aufgebaut:
 - Sichtung vorliegender Dokumentation
 - klärende Gespräche mit Benutzern und Entwicklern
 - Abgleich der Funktionalität
 - Dokumentation und erste Auswertung der Checkliste

Eine Grundorientierung lässt im Allgemeinen schon ein Urteil zu, ob die zur Auswahl anstehenden Systeme sich überhaupt für den vorgesehenen Zweck eignen

Softwarauswahl / -inspektion (1)

- Überprüfung der Checkliste
 - Checkliste wird durch intensive Gespräche mit den Benutzern verifiziert
 - Erhebungsbogen des abzulösenden Systems wird nach Darstellungsform und Darstellungstiefe überprüft, ob es eine einigermaßen objektive Abbildung des realen Systems ermöglicht
 - Zusätzliche Überprüfung der Relevanz der angesprochenen Checklistenpunkte aus der Sicht des späteren Benutzers
 - Häufig ergeben sich zusätzliche Gesichtspunkte, die für eine Beurteilung wichtig sein können

Softwareauswahl / -inspektion (2)

- Besichtigung einer Systemrealisierung reicht vielfach nicht aus
- Mindestens ein Benutzer des Alt-Systems sollte mit einem Benutzer des Neu-Systems unter Demonstration und Diskussion der Anwendermasken und Listoutput weitgehende Entsprechung der Analysepunkte feststellen und dokumentieren
- Persönliche Eindruck der Beurteiler ist festzuhalten
- Beurteilung durch mindestens zwei Benutzer (oder auch Entwickler); ihnen sollte genug Zeit eingeräumt werden

Softwareauswahl / -inspektion (3)

- Abhängig von der Qualität der Softwaredokumentation kann die Analyse sehr weit getrieben werden
- Das Ergebnis ist eine scheinbare Objektivität, bei der es sich um eine Meinungsbildung von fachkundigen Individuen handelt
- Aussagen zur Qualität können eigentlich nur lauten: "... das untersuchte System B erscheint (in hohem oder weniger hohem Maße) geeignet zu sein, das frühere System A zu ersetzen ..."
- Zahlenmäßige Fixierung in Form von "...entspricht zu 80% der bisherigen Lösung ..." sind nur schwer möglich, da Systemphilosophien und Softwarerealisierungen vielfach zu unterschiedlich sind
- Versuche, Urteile auf quantifizierbare Fakten zu gründen sind somit äußerst schwierig

Softwareauswahl / -inspektion (4)

- Beispiel-1: Angaben über LOC (Lines of Codes)
 - wird genutzt, um Umfang der Programme und damit anscheinend auch ihre Komplexität zu fixieren
 - jedoch fragwürdig, da schon seit geraumer Zeit viele Programmteile durch Generatoren automatisch erzeugt werden
- Beispiel-2: Anzahl der Programm-Module
 - nur von begrenztem Wert, da eine kritische Überprüfung der "Modularität", der "Schnittstellen" und der "Systemstruktur" häufig nicht möglich ist
 - Im günstigsten Fall gewinnt man Einsichten über Systemzusammenhänge (was aber auch schon wertvoll ist!)

Softwareauswahl / -inspektion (5)

- Beispiel-3: investierte „Manpower“

Angaben über „Manpower“, die in ein Anwendersoftwarepaket (angeblich) investiert wurde, kann schwerlich zur Quantifizierung herangezogen werden, da

 - z.B. die Produktivität der Entwickler sowohl personenabhängig wie auch firmenspezifisch ist
 - z.B. durch Verwendung von 4ten Generationssprachen so erheblich gesteigert werden kann, dass solche Angaben ebenfalls irreführend wären

Spezifizierung neuer Anforderungen (1)

- Bei der Analyse von Softwarelösungen fehlen oftmals Funktionalitäten, auf die bei einer zukünftigen Lösung Wert gelegt wird.

Solche "features" lassen sich in folgende Gruppen einteilen:

- bei der neuen Lösung bereits vorgesehen
- bei einer neuen Lösung durch den Anwender leicht zu realisieren (ohne Eingriffe in die Systemstruktur)
- solche, auf die man im Interesse eines einheitlichen Produktes besser verzichtet

Spezifizierung neuer Anforderungen (2)

- Softwarehersteller sind bei Erschließung neuer Marktsegmente an Kooperationen mit Anwendern interessiert. Folgende Fälle werden betrachtet:
 - Anwender hat sehr klare, durch das bisherige Produkt geformte Vorstellungen, die auch durch formale Beschreibungen fixierbar sind
 - der Anwender hat nur ungefähre Vorstellungen
- Standardisierung bei unklaren Vorgaben ist anzustreben, mit dem sich dann der Anwender begnügen muss
- Professionelle Anbieter haben firmenspezifischen Standard (SAP, Microsoft) als Basis ihrer Entwicklung aufgebaut und dokumentiert

Maßnahmen zur Absicherung der Auswahl

- Beschaffung von Softwaresystemen hat für betroffene Anwenderorganisation großer Bedeutung
 - Einfluss auf gesamte Ablauforganisation
 - Erhebliche Aufwendungen für Schulung und Betreuung
 - Übergang von einer vertrauten Lösung auf eine neue; oft mit nicht vorhersehbaren Schwierigkeiten verbunden
 - Hinterfragung beim Anbieter, wo Schwierigkeiten erfahrungsgemäß auftauchen, um Vorkehrungen der Risikominimierung zu treffen

Aufbau eines professionellen Projektmanagement- und Qualitätssicherungs-Verfahrens, dass bis zur Abnahme mitgeführt und aktualisiert wird