

Standard Software

Inhaltsverzeichnis:

<i>Definition</i>	2
<i>Merkmale</i>	2
<i>Alternative zu SASW</i>	2
<i>Arten</i>	3
<i>Voraussetzungen</i>	3
<i><u>Make or Buy</u>: Gründe für SASW</i>	4
<i><u>Make or Buy</u>: Gründe gegen SASW</i>	4
<i><u>Make or Buy</u>: Zwänge zur Individualentwicklung</i>	4
<i><u>Make or Buy</u>: Vorteile von Individualentwicklung</i>	4
<i>Ziele</i>	5
<i>Entscheidungskriterien</i>	5
<i>Kritische Erfolgsfaktoren</i>	6
<i>Grenzen</i>	6
<i>Chancen</i>	6
<i>Risiken</i>	6
<i>Massnahmen</i>	6
<i>Anpassungsmöglichkeiten von SASW</i>	7
<i>Qualitätsmerkmale von SASW</i>	7
<i>Analyse der SASW-Integration</i>	7
<i>Vorgehen beim Auswahl, Einführung von SASW</i>	8
<i>Kosten</i>	9
<i>Vorteile</i>	9
<i>Nachteile</i>	9
<i>Produktbeispiele</i>	10
<i>Standardsoftware als spezifisches Projektvorgehen</i>	10

Standard Software

Definition

- Programme, die von grosser Zahl Anwender zur Lösung gleicher oder ähnlicher Aufgabenstellung eingesetzt werden können.
-

Merkmale

- Nur für bestimmten Anwendungsbereich
SASW deckt einen genau definierte Funktionsumfang ab
 - Keine spezifische Anforderungen
unternehmensspezifische Anforderungen werden im „Standard“ nicht angeboten
 - in mehreren Unternehmen einsetzbar
 - Einführung mit geringem zeitlichen, finanziellen Aufwand
 - Voraus bekannten Preis
Einschliesslich Beratungs- und Serviceleistungen
 - Abhängigkeit
Standard-SW macht abhängig, bindet das Unternehmen an den SW-Lieferanten und dessen betriebswirtschaftliche Entwicklung
 - Kultureller Wandel
Wechsel von Eigenentwicklung zur Standard-SW bedeutet für Informatik-Entwicklung einen kulturellen Wandel
 - gezieltes Erneuern der SW
Einkauf neuer „Release“ → Standard-SW gezielt modernisieren
Aktualisieren ist periodisch (hingegen Individual-SW ist laufend)
-

Alternative zu SASW

- Eigenentwicklung (Individual-SW)
 - Fremdbezug von Informatik-Dienstleistungen (Outsourcing)
 - Konzeptübernahme (Zukauf Lösung eines Marktpartners)
 - Entwicklungskooperation von Unternehmen gleicher Branche
 - Zukauf von Standard-SW
-

Standard Software

Arten

- Standard-Anwendungs-SW:
 - Finanz- / Rechnungswesen
 - Personalwesen
 - Marketing
 - Materialwesen
 - Betrieb-SW:
 - Betriebssysteme
 - Programmiersprachen
 - Utilities
 - Datenbankmanagement-Systeme
 - Kommunikations-SW
 - Tools:
 - Textverarbeitungs-SW
 - Tabellenkalkulations-SW
 - Auswertungs-SW (QMF, SQL, SESAM, Nomad)
 - Statistik-SW (FOCUS)
 - Grafik-SW
-

Voraussetzungen

- Konsequenter Einsatz einer Informatik-Strategie:
 - Technologie-Management
 - Beobachtung der Technologie-Entwicklung
 - Bestimmung des Technologie-Bedarfes
 - Systematisches Vorgehen bei Projekten
 - Klare leistungsgebunden vertrag mit SASW-Lieferant, Sicherstellen der Wartung
 - Sauberer Schnittstellen zur Funktions- und Datenbasis bestehender Software
 - Genügende Daikennnisse über das Software-Produkt
 - Evaluation mit klaren Spezifikationen (Pflichtenheft)
-

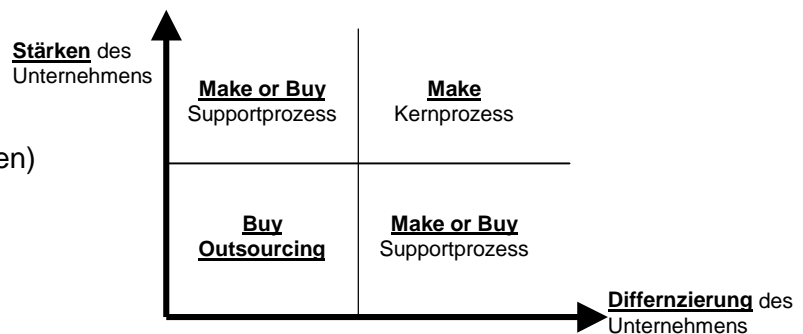
Make or Buy

Matrix:

Kernprozesse:

(abgeleitet aus Kernkompetenzen)

1. Produktentwicklung
2. Produkte an Kunden liefern
3. Kundenbeziehung pflegen
4. ...



Standard Software

Make or Buy: Gründe für SASW

- Kostenüberlegungen
 - Mangel an finanziellen Mitteln
 - Fehlende Entwicklungszeit
 - Fehlende Infrastruktur
 - Personelle Gründe
 - Mangelndes Know-how
 - IS-Strategie/-Konzept
 - Anwendungsstau in eigenen EDV-Entwicklung
 - Integration von betrieblichen Funktionsbereichen (SAP-HR, PK)
 - Internationale Verflechtung
 - Schrittweise Einführung / Ausbau ist möglich
-

Make or Buy: Gründe gegen SASW

- Unabhängigkeit des Unternehmens
 - Wettbewerbsvorteile durch eigene Speziallösung
 - Falls keine ausgereifte SASW auf dem Markt gibt
 - SASW-Anbieter wird bald vom Markt verschwinden
 - HW-Konfiguration ist zu klein oder zu gross dimensioniert
 - SASW passt der HW-Architektur des Unternehmens nicht
 - SASW kann nicht konzernweit eingesetzt werden
-

Make or Buy: Zwänge zur Individualentwicklung

- Vorhandene Datenbasiselemente, die nicht zur SASW passen
 - „exotische“ HW-/SW-Umgebung
 - kein genügendes Marktangebot von SASW
-

Make or Buy: Vorteile von Individualentwicklung

- Entwicklungs-Know-how bleibt im Hause
 - Massgeschneiderte Lösungen sind möglich
 - Punktuelle Verbesserungen sind möglich
 - Kleinere externe Wartungsabhängigkeit
 - Weniger SW-technische Konflikte (alles aus einem Guss)
 - Möglichkeit für Einsatz moderner Realisierungstechnik
 - Freiheit in der Schnittstellengestaltung
-

Standard Software

Ziele

- Operative Ziele:
 - Beschleunigung der Bearbeitung
 - Weniger Papierbearbeitung
 - Bessere Kommunikation aller Mitarbeiter
 - Höhere Sicherheit bei Informationsspeicherung
 - Schnelleres Wiederfinden von Informationen

 - dispositive Ziele:
 - Bessere Kontrolle von Entscheidungen
 - Einsatzsteuerung von Mitarbeitern
 - Erfolgskontrolle einzelner Produkte

 - Strategische Ziele:
 - Aufbau eines unternehmensweiten Informationssystems (IS)
 - Erlangung von Wettbewerbsvorteilen durch Nutzung moderner Kommunikations- und Informationssysteme
 - Erhöhung der Kundenzufriedenheit
-

Entscheidungskriterien

- Integration:
 - Mindestkompatibilität zu Basisdatensystem
 - Anpassungsaufwand der vorhandenen Systeme
 - Operationsrisiko bei Änderungen

 - Funktionalität
 - Ob KO-Kriterien (MUSS-Kriterien) erfüllt sind
 - Handhabungskomfort: Kann Käufer selbst parametrisieren?
 - Einsparungsgrades in Informatikstrategie

 - Qualität:
 - Mit welchen Methoden, Werkzeugen wurde SASW entwickelt?
 - Lieferantenbeurteilung
 - Klare Definition von SW-Bausteinen
 - Ist automatische Anpassung an HW-/SW-Umgebung möglich?
 - Dokumentation: Welche Sprachen, Detailierungsgrad?
 - Entwicklungsfähigkeit
-

Standard Software

Kritische Erfolgsfaktoren

- Mut zum Entscheid (keine Hemmung vor Fremd-SW)
 - Erfüllung technologische Anforderungen bei übergreifenden Systemen,
 - Genügende Kenntnisse von SW-Produkt
-

Grenzen

- Wenn Wettbewerbsvorteile nur durch eigene Speziallösung (Individual-SW) erreicht werden können, dann auf keinen Fall SASW einsetzen.
-

Chancen

- Eine Lösung für personelle Engpässe
 - Bessere Qualität mit weniger Aufwand
 - Rascher Einsatz der SASW
-

Risiken

- Mangelnde Detailkenntnisse der SASW bei Anwender
 - Ängste vor Fehlentscheidungen
 - Fehlende/ mangelhafte Anforderungskatalog
 - Subjektive Bewertung beim Auswahl
-

Massnahmen

- Systematisches Vorgehen (... , Evaluation, Einführung, ...)
 - Anwender früh beiziehen: Pflichtenheft
 - Offene Kommunikation über das Vorhaben
 - Schulungsplan der SASW
-

Standard Software

Anpassungsmöglichkeiten von SASW

- Tabellen Kategorie 1:
SASW wird für HW-/SW-Umgebung durch SASW-Käufer parametrisiert
 - Tabellen Kategorie 2:
SASW wird für unternehmensspezifische Abläufe, Masken, Wertemengen durch SASW-Käufer parametrisiert
 - Tabellen Kategorie 3:
Tabellen, die nur mit Rücksprache mit SASW-Hersteller geändert werden dürfen. Bei Fehlerfall wird die Haftung abgelehnt. Diese Tabellen verursachen grösserer Aufwand bei Release-Wechsel.
-

Qualitätsmerkmale von SASW

- Allgemeine Merkmale:
Effizienz: Zeiteffizient, Speichereffizient
Flexibilität: Kompabilität, Portabilität, Adaptabilität
Zuverlässigkeit: Fehlerfreiheit, Ausfallsicherheit
Benutzerfreundlichkeit
 - Funktionale Merkmale:
Funktionsumfang
Funktionsqualität
-

Analyse der SASW-Integration

- Wie wird SASW in bestehende IS-Architektur integriert ?
 - Kann die SASW in das unternehmensweite Datenmodell integriert werden?
 - Wie ist das Vorgehen bei der Integration der SASW in das unternehmensweite Datenmodell ?
 - Nutzen des Datenmodells: Können Synonym-/Homonym-Problematik überprüft werden?
 - Physische Datenintegration: Schnittstellenarchitektur für gezielte Abfrage von Informationen durch die Umsysteme. bzw. Bilden von Extrakte Daten.
 - Nutzenpotential der Integration:
 - Nutzung gemeinsamer Datenbestände
 - Transparente Datenplattform für Auswertungen
 - Einfache, transparente Integration neuer Systeme
 - Bessere Akzeptanz bei den Anwendern
 - Reduktion des Schulungsaufwands
 - Reduktion des Betriebsaufwands
-

Standard Software

Vorgehen beim Auswahl, Einführung von SASW

- Projektstart:
Ziele und Umfang der neuen SW definieren → Projektauftrag
- Ist-Analyse:
Arbeitsabläufe, Aufgaben, Informationsflüsse, Belege, Formulare, ausführende Stellen in einem begrenzten Bereich analysieren → Stärken/Schwächen
- Konzepterstellung:
Erarbeiten eines Konzepts als Rahmenvorschlag ohne konkrete Entscheidung → Grobkonzept
- Anforderungsdefinition:
Betriebswirtschaftliche, technische Anforderungen an die spätere SW (fachliche, HW-Anforderungen) → Pflichtenheft
- Markterhebung/Offerten einholen:
Untersuchen von Fachpresse, SW-Kataloge (Bsp. ISIS Software Report), Messenbesuch (Bsp. CeBIT), Fachvorträge, Referenzkunden, SW-Hersteller, Unternehmensberatungen, unabhängige SW-Unternehmen, Internet, → Marktstudie
- Bewertung/Auswahl:
Erstellen Bewertungsdokumente, Kosen/Nutzen-Gegenüberstellung, Wirtschaftlichkeitsberechnung, Auswahl, Entscheid → Kriterienkatalog, Bewertungsliste, Evaluationsbericht, Kosten-Nutzen-Analyse, Antrag an GL
- SW-Test:
SW auf KO-Kriterien prüfen, SW anhand der Kriterien im Pflichtenheft untersuchen → Testbericht
- Pilotinstallation:
Neue SW vor Beginn des Migrations, Echtbetriebs 2 Monate lang parallel mit dem alten SW einsetzen.
- Entscheid:
Präsentation für GL (speziell der Management Summary-Teil)
- Vertragsabschluss
Vertrag umfasst Kaufpreis, Schulungs-, Beratungsleistungen, SW-Anpassungen, spätere Wartung, Termine, Nutzungsrechte, Haftung der Anbieter → Nutzungsvertrag
- Einführung/Schulung:
Die HW-Installation in der Fachabteilung abschliessen, Migration der alten Daten, Pilot-Betrieb, Muster-Daten für Schulung vorbereiten, Schulungsplan mit Einführungsplan koordinieren → Schulungskonzept, Einführungsplan

Standard Software

Kosten

- Lizenzpreis
 - Hardwarekosten
 - Wartungsgebühr (wiederkehrend)
 - Parametrisierungsaufwand (Customizing)
 - Installations-/Anpassungsaufwand
 - Releasewechsel-Aufwand
 - Integrations-/Schnittstellen-Aufwand
 - Reorganisations-Aufwand
 - Schulungs-/Beratungsaufwand
 - Evaluationsaufwand:
 - IST-Analyse
 - Reisen und Seminarbesuche
 - Beratung, Umstellunterstützung
 - Installation
-

Vorteile

- Wirtschaftlichkeit (Eigenentwicklung ist 10-50 mal teurer)
 - Zeitvorteil (keine Entwicklungszeit bei Anwender)
 - Schnelle Bewältigung der Probleme (Kaufen und einsetzen)
 - Qualität (vielen Kunden → Fehler werden schnell entdeckt)
 - Ergonomie
 - Dokumentation
 - Vermeiden von Personalengpass beim Anwender
 - Organisatorische Lösung wird mitgeliefert (Ablauforganisation)
 - Wartungsaufwand lässt sich genau kalkulieren
-

Nachteile

- Anpassungsprobleme: keine unternehmensspezifische Fkts.
 - Abhängigkeit von Lieferanten/Hersteller
 - Inegration/Schnittstellenprobleme
 - „mitgelieferte“ Ablauforganisation passt nicht immer
 - Wenn Qualität der SASW nicht den Anforderungen entspricht
 - Umfangreiche Funktionalitäten, die nicht alle erforderlich sind
 - Überproportionale Ressourcenverbrauch von HW
 - Zwang zum periodischen Release-Wechsel
 - Fehlende Kompatibilität zu den Umsystem oder Altsystem
-

Standard Software

Produktbeispiele

Produktenamen	Anbieter
Navision	Navision
JD Edwards	JD Edwards
R3 R2	SAP
Pius Penta	PSI
=Office 2=	Informing
Baan	Baan
Steeb	Steeb
MMS	ProByt

Standardsoftware als spezifisches Projektvorgehen

Standardsoftware

Standardsoftware vs. Eigenentwicklung

+	Die Wirtschaftlichkeit ist Vorteil Nummer eins. Individualsoftware (Eigenentwicklung) ist im Entwicklungspreis zehn- bis fünfzigmal teurer als Standardsoftware
+	Die Qualität ist oft besser
+	Ergonomie und Dokumentation der Programme sind im Gegensatz zu denjenigen der Individualsoftware meistens sehr gut
+	Es entsteht ein Zeitvorteil, da die Entwicklung nicht beim/mit dem Anwender erfolgt.
+	Schnelle Umsetzung der Probleme. Da die Standardsoftware auf dem Markt erhältlich ist, kann diese ohne Verzug gekauft und installiert werden.
+	Ein Personalengpass beim Anwender wird vermieden. Die Einstellung von Softwarespezialisten erübrigt sich
+	Hinter Standardsoftware steht eine organisatorische Lösung. Besonders kleinere Unternehmen schätzen den Umstand, dass ihnen eine zweckmässige Ablauforganisation quasi mitgeliefert wird. Mit dieser organisatorischen Lösung besteht generell die Möglichkeit „alte Zöpfe“ abzuschneiden, d.h. die organisatorischen Abläufe können überarbeitet werden.
+	Mit Standardsoftware kann der Wartungsaufwand relativ genau kalkuliert werden. Erweiterungen (zusätzliche Funktionalität) können gezielt eingekauft werden.
+	Einfachere Kalkulation durch voraussehbare Kosten
-	Anpassprobleme: Nicht allen Wünschen des Anwenders kann entsprochen werden. Die SASW löst Standardprobleme, und manche Funktionen sind nicht vorgesehen.
-	Abhängigkeit vom Lieferanten/Herstellen: Oft ist es dem Anwender unmöglich, Fehler selbst zu beheben oder Anpassungen an neue Gegebenheiten vorzunehmen. Ausserdem ist der Anwender auf dessen Marktpräsenz angewiesen
-	Integration/Schnittstellenprobleme: Standardsoftware lässt sich oft nicht in die Gesamtinformatik des Unternehmens integrieren. In diesen Fällen müssen aufwendige Schnittstellen entwickelt und unterhalten werden.
-	Die „mitgelieferte“ Ablauforganisation kann sich dort als Nachteil erweisen, wo Arbeitsabläufe gut geregelt sind oder sich nicht ohne weiteres ändern lassen.
-	Die Qualität der SASW entspricht nicht den Qualitätsanforderungen des Unternehmens. Beispiel: alte Batchverarbeitungsprogramme wurden durch Bildschirm-Ein- und Ausgaben ergänzt und damit nur äusserlich modernisiert.

Standard Software

-	Die unternehmensweite angestrebte IS-Architektur, wird nicht unerstützt oder sogar verhindert; d.h. es entstehen neue Inseln.
-	Zu umfangreiche (mitgelieferte) Funktionalität. Es werden Funktionen mitgeliefert, die eigentlich nicht erforderlich wären.
-	Ueberproportionale/r Ressourcenverbrauch resp.-belastung (Hardware).
-	Zwang zum periodischen Releasewechsel. Nur somit kann der Anwender von den Fehlerkorrekturen profitieren.
-	Fehlende Kompatibilität zu den Um- und/oder Altsystemen.

Standardsoftware SASW "Kosten"

•	Lizenzpreis
•	Wartungsgebühr
•	Parametrisierungs-Aufwand
•	Installations-/Anpassungsaufwand
•	Releasewechsel-Aufwand
•	Integrations-/Schnittstellen-Aufwand
•	Reorganisations-Aufwand
•	Schulungs-/Beratungsaufwand
•	Evaluationsaufwand
•	Hardwarekosten

Pflichtenheft

Vor-Nachteile

+	Gesprächsbasis zwischen Kunde und Lieferant
+	Klare Definition des Funktionsumfangs
+	Man muss sich frühzeitig mit den Problemen auseinandersetzen
+	Transparentere selektion der Anbieter
+	Kann als Vertragsgrundlage beigezogen werden
+	Macht eine Evaluation transparent
-	Sehr aufwendig zu erstellen
-	Kann bei zu spät entdeckten Anforderunge einschränkend wirken.

Systemanbieter

	Produktenamen	Anbieter
•	Navision	Navision
•	JD Edwards	JD Edwards
•	R3 R2	SAP
•	Pius Penta	PSI
•	=Office 2=	Informing
•	Baan	Baan
•	Steeb	Steeb
•	MMS	ProByt

Standard Software

Wirtschaftlichkeit von Modifikationen

Anpassungskategorien

Kategorie	Beschreibung der Anpassungen
Kat 1	<p>Basisparametrierung</p> <p>Damit die SASW grundsätzlich in der HW- / SW-Umgebung des Anwenders lauffähig ist</p> <ul style="list-style-type: none"> • Betriebssystem • Datenbanksystem • Netzwerkumgebung (Drucker, Emulationen, usw) • TP-Monitore (CICS, IMS)
Kat 2	<p>Applikationsparametrierung</p> <ul style="list-style-type: none"> • Masken (Abfolge) • Felder (Muss, Kann, unsichtbar) • Abläufe (Steuerung des Werte- und Mengenflusses) • Wertemengen
Kat 3	<p>Code-Modifizierung</p> <p>Tabellen und Programme die nicht ohne Rücksprache mit dem Hersteller geändert werden dürften</p> <ul style="list-style-type: none"> • Tabellen Struktur • Attribute (löschen , hinzufügen, usw.) • Felder (Längen, Typen (Char, Num, usw)) • Programme

Anpassungskosten

Anpassungen in	Auswirkungen
Kat 1	<ul style="list-style-type: none"> • Keine negativen • Kostengünstig • Schnell • einfach
Kat 2	<p>Keine negativen</p> <ul style="list-style-type: none"> • Kostengünstig • Schnell • einfach
Kat 3	<p>Nur negative</p> <ul style="list-style-type: none"> • Erhöhter Aufwand bei Releasewechsel (höhere Kosten) • Support vom Hersteller nicht mehr gewährleistet • lange Wartezeiten um Fehler abzuklären • Eigene Entwickler notwendig (Teuer / Abhängigkeit)