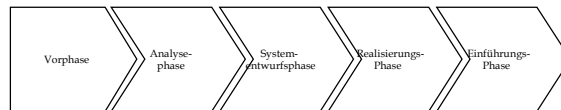
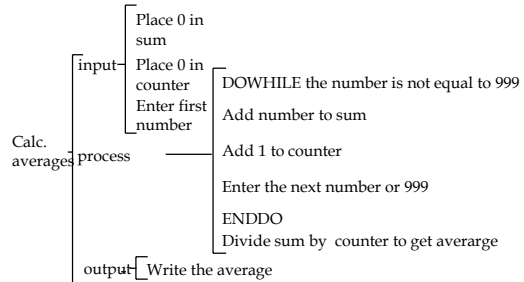
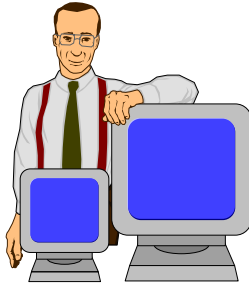


2.5. Systementwicklung



2.5. Systementwicklung

Ziel:

Darstellung der grundlegenden Vorgehensweise der Entwicklung von betrieblichen Anwendungssystemen in Projektform

Gliederung:

1. Begrifflichkeiten
2. Zentrale vs. dezentrale Systementwicklung
3. Entwicklung von Anwendungssoftware

Literatur:

Stahlknecht/Hasenkamp S. 226 - 341
 Hansen S. 134 - 167

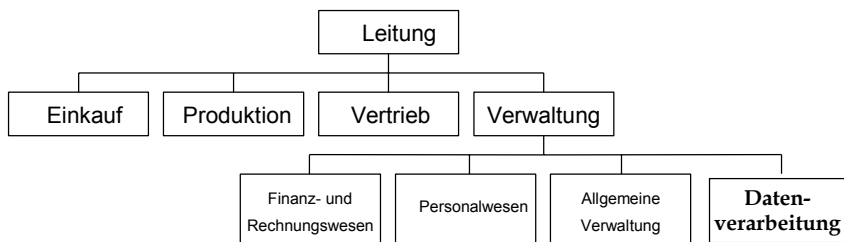
1. Begrifflichkeiten Anwendungssysteme

Die Systementwicklung bezieht sich hier nur auf Anwendungssysteme. Ein Anwendungssystem ist eine Software für ein konkretes betriebliches Anwendungsgebiet entwickelt und eingeführt wird (Dispositionssysteme für die Lagerhaltung, Fakturierungsprogramme, Gehaltsabrechnungssysteme).

Heutige Systementwicklungen orientieren sich häufig nicht mehr an den betrieblichen Funktionen (Beschaffung, Produktion, Absatz) sondern an Geschäftsprozessen, die durch alle oder mehrere Funktionen eines Betriebes laufen.

Die Vorgehensweise einer solchen Systementwicklung entspricht fast immer dem Phasenmodell, während das Projekt häufig in Form einer Matrix organisiert ist.

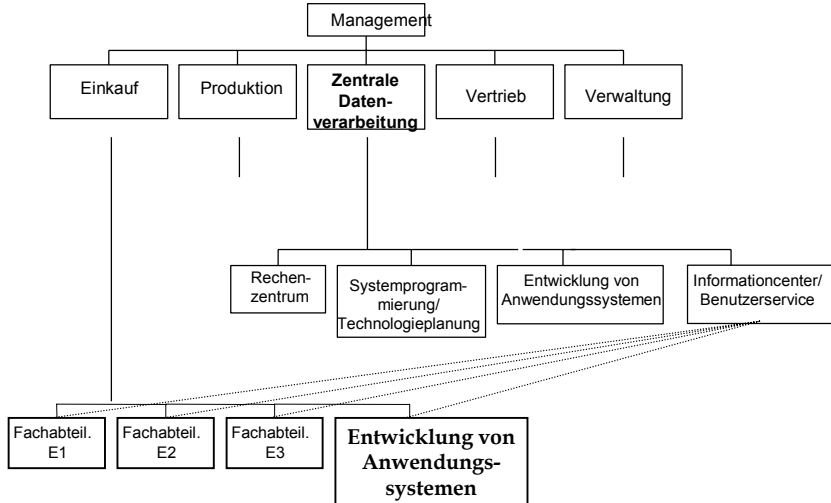
2. Zentrale vs. dezentrale Systementwicklung Klassische organisatorische Struktur



Unterstützung ggf.:

- o Stabsbereiche (Planung, Int. Revision)
- o Servicebereiche (DV-, Rechtsabteilung)

2. Zentrale vs. dezentrale Systementwicklung Neuere organisatorische Struktur



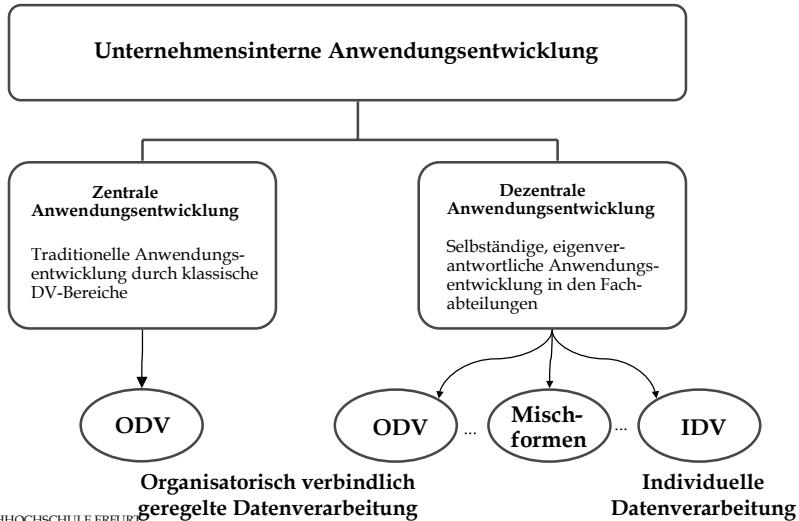
2. Zentrale vs. dezentrale Systementwicklung Dezentrale Anwendungsentwicklung, warum?

- Sichtbarer Anwendungsstau in deutschen Unternehmen von 2 - 5 Jahren (nicht sichtbarer Anwendungsstau 3 - 8 Jahre)
- Komplexität der Anwendung wird allein vom Endbenutzer in allen Einzelheiten erfasst
- Werkzeuge sind heute so komfortabel geworden, daß eine Programmierung im eigentlichen Sinne häufig nicht mehr erforderlich ist (Modellierung)
- Endbenutzer sind wesentlich kompetenter geworden
- Die Art der zu entwickelnden Systeme lassen eine zeitintensive zentrale Lösung gar nicht mehr zu
- Die technischen Möglichkeiten lassen eine Dezentralisierung zu. Fast jeder Arbeitsplatz hat heute eine eigenen PC, der mittels Netzwerk mit anderen IV-Ressourcen verbunden ist.

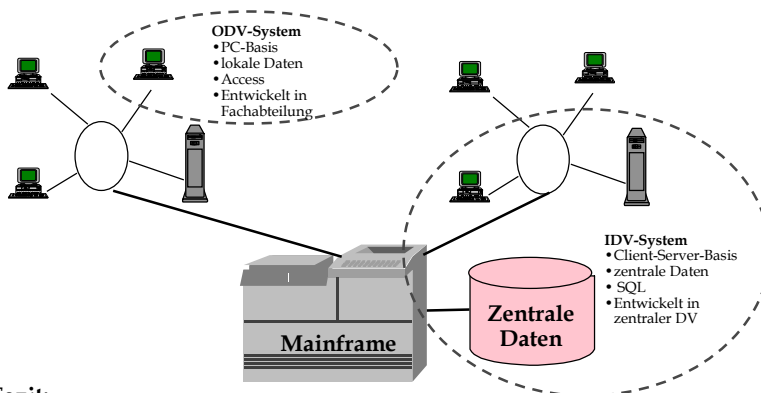
Fazit:

Die Dezentrale Anwendungsentwicklung ist in den meisten Unternehmen erklärte *Informatikstrategie* geworden

2. Zentrale vs. dezentrale Anwendungsentwicklung Auflösung des Zielkonflikts



2. Zentrale vs. dezentrale Anwendungsentwicklung Keine Frage der Hardware



Fazit:
IDV oder ODV ist keinesfalls abhängig von der Art der Rechner, dem Ort der Datenhaltung, den eingesetzten Werkzeugen oder dem Ort der Entwicklung.

2. Zentrale vs. dezentrale Anwendungsentwicklung Gestaltungs- und Benutzungsspielräume

Was ist eigentlich

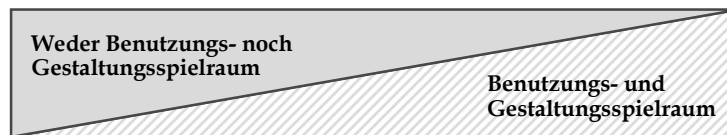
Gestaltungsspielraum?

Mitarbeiter besitzt die Freiheit, seine Systeme individuell zu gestalten. Keine Einschränkungen der Systemgestaltung durch datenschutzrechtliche, arbeitsrechtliche oder sonstige Vorschriften.

Benutzungsspielraum?

Mitarbeiter besitzt die Freiheit, ob, wann und in welchem Umfang er die eigenerstellten Systeme einsetzt.

2. Zentrale vs. dezentrale Anwendungsentwicklung Abgrenzung IDV und ODV



ODV-System

Systeme zur externen Rechnungslegung

- strenge gesetzl. Restriktionen
- revisionsfähig

Mischformen

Systeme für die Analyse von Planabweichungen

- bestimmte innerbetr. Restriktionen
- revisionsfähig

IDV-System

Ad-hoc-Systeme zur Investitionsbeurteilung

- keine Restriktionen
- persönliche Ergebnisverantwortung

2. Zentrale vs. dezentrale Anwendungsentwicklung Koordination durch Information Centers

Definition:

Information Center sind formal in die Organisation eingebettete Stellen oder Abteilungen, die sich mit sämtlichen Fragestellungen beschäftigen, die durch den Einsatz der Dezentralen Anwendungsentwicklung tangiert werden.

Aufgabengebiete

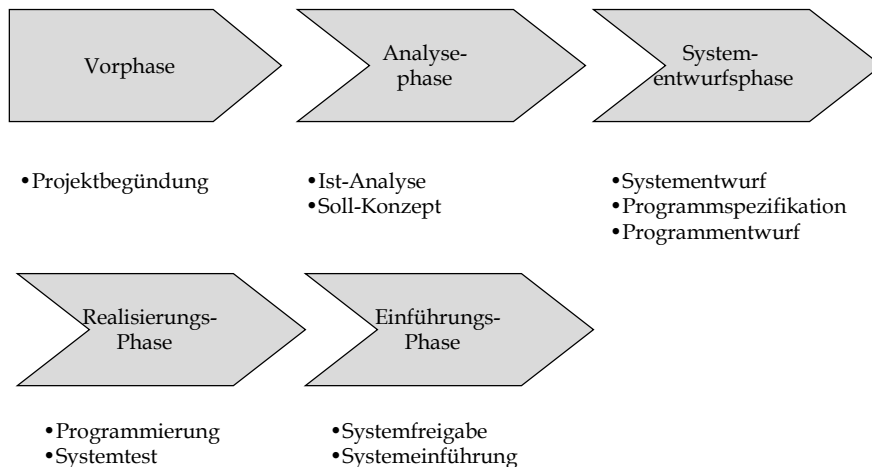
- Unterstützung der Benutzer bei der Entwicklung
- Durchführung von Schulungen
- Standardisierung eingesetzter Hard- und Software
- Standardisierung von Vorgehensweisen bei der Entwicklung
- Steuerung & Kontrolle des Datenaustausches
- Erteilung & Verwaltung von Zugriffsberechtigungen
- Beurteilung von Benutzer-Vorhaben (IDV, ODV)
- Förderung des Erfahrungsaustausches

P.S.

Das erste Information Center wurde bei der IBM-Kanada aufgebaut



3. Entwicklung von Anwendungssoftware Phaseneinteilung eines IV-Projekts



3. Entwicklung von Anwendungssoftware

Projektbegündung

In der Phase Projektbegündung (Preliminary Investigation) ist es wichtig, sich über die drei folgenden Punkte Klarheit zu verschaffen:

Natur des Problems:

Was ist das Problem?

Was für grundsätzliche Möglichkeiten existieren?

Umfang

Welcher Problemumfang wird exakt bearbeitet?

Anforderungen

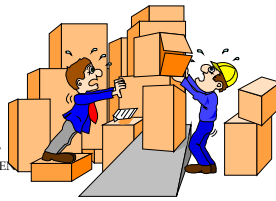
Was soll das System exakt können?

Projektvorschläge
Projekterwartungen

Managementmitteilung

Vorstellung und Empfehlungen hinsichtlich der Ergebnisse der Phase

- (1) Problem wird als nicht relevant erkannt
- (2) Problem wird direkt gelöst
- (3) Aufforderung, die nächste Phase zu eröffnen und detailliertere Analysen zu betreiben

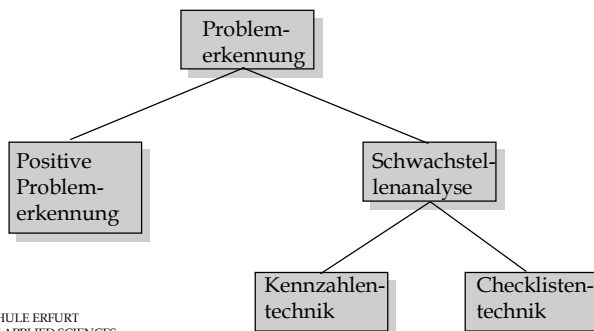


3. Entwicklung von Anwendungssoftware

Problemerkennung

Probleme werden häufig erst erkannt, wenn ihre Auswirkungen bereits so gravierend sind, dass sie nicht mehr übersehen werden können oder die Entdeckung ist abhängig vom Zufall.

Die folgenden Methoden können zur Problemerkennung verwendet werden:



3. Entwicklung von Anwendungssoftware

IST-Analyse

Ziel:

Mit der IST-Analyse sollen die *Schwachstellen* der bestehenden Ablauforganisation herausgearbeitet und somit die Voraussetzungen für die Aufstellung eines SOLL-Konzeptes geschaffen werden.

Erhebung/ Beschreibung des IST-Zustands

- Arbeitsabläufe/ Geschäftsprozesse mit zeitlichem Verlauf und beteiligten Stellen
- Entstehung, Verwendung und Mengengerüst aller relevanten Daten
- Schnittstellen zu internen und externen Stellen
- Kosten des Ist-Zustandes

Bewertung des IST-Zustands

- Schwachstellen ermitteln (z.B. Medienbrüche, Inflexibilitäten)
- Folgeschäden abschätzen (z.B. Mehraufwand, entgangenes Geschäft)

3. Entwicklung von Anwendungssoftware

IST-Analyse/Erhebungstechniken

Dokumentenanalyse

- Geringer Zeitaufwand
- Keine Störungen des Ablaufs
- Gefahr fehlender Aktualität
- Zur Einarbeitung in Problemstellung
- Kontrolle von Informationen

Fragebogen

- Geringer Zeit- und Kostenaufwand
- Gute Auswertungsmöglichkeiten
- Gefahr von Missverständnissen
- Gefahr der Manipulation
- Zur groben Erfassung von Tatbeständen
- Bei grosser Anzahl der zu Befragenden

Interview

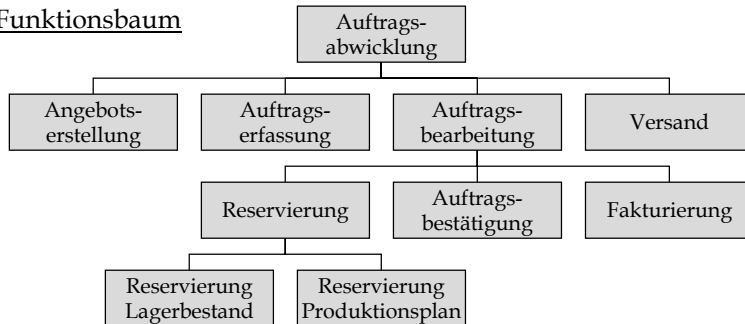
- Direkte Beteiligung der Betroffenen
- Flexible Gestaltung
- Objektivität fraglich
- Hoher Zeitaufwand
- Zur Erhebung qualitativer Einflussgrößen

Beobachtung

- Geringe Störung des Betriebsablaufs
- Relativ hoher Zeitaufwand
- Zur Überprüfung von Tatbeständen
- Zur Erkennung von Engpässen

3. Entwicklung von Anwendungssoftware IST-Analyse/Techniken der Beschreibung

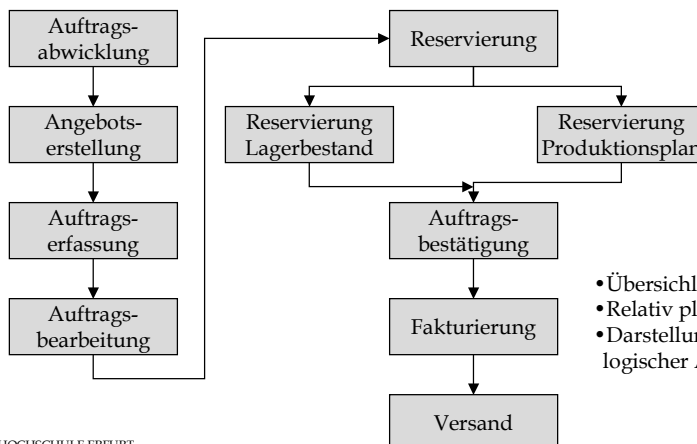
Funktionsbaum



- Reduktion der Komplexität
- Nur statische Darstellung

3. Entwicklung von Anwendungssoftware IST-Analyse/Techniken der Beschreibung

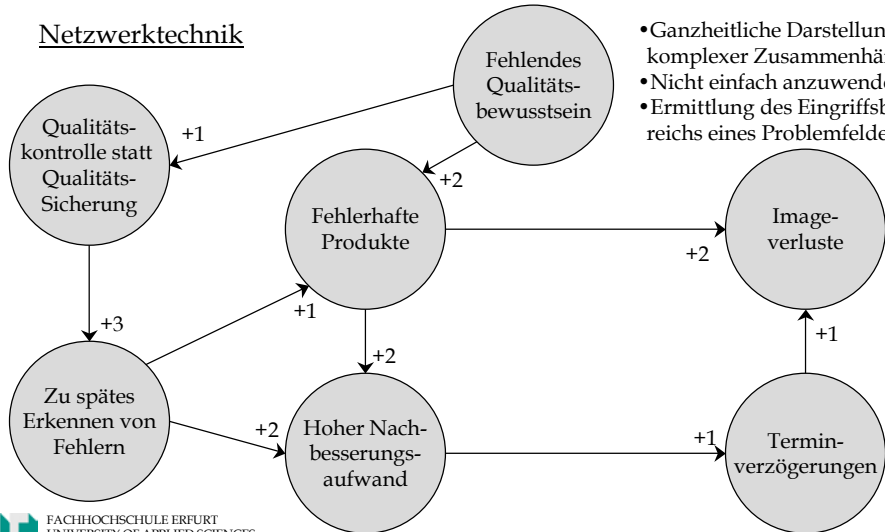
Ablaufplan



- Übersichtliche Darstellung
- Relativ platzaufwendig
- Darstellung zeitlicher u. logischer Abläufe

3. Entwicklung von Anwendungssoftware IST-Analyse/Techniken der Analyse

Netzwerktechnik

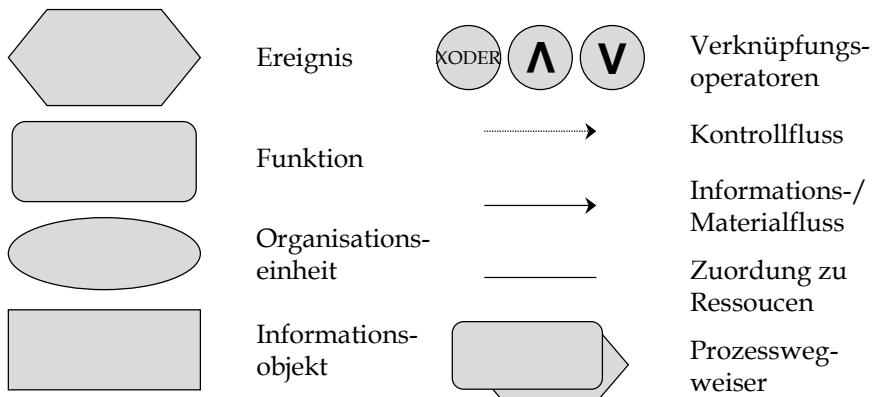


- Ganzheitliche Darstellung komplexer Zusammenhänge
- Nicht einfach anzuwenden
- Ermittlung des Eingriffsbereichs eines Problemfeldes

3. Entwicklung von Anwendungssoftware IST-Analyse/Techniken der Analyse

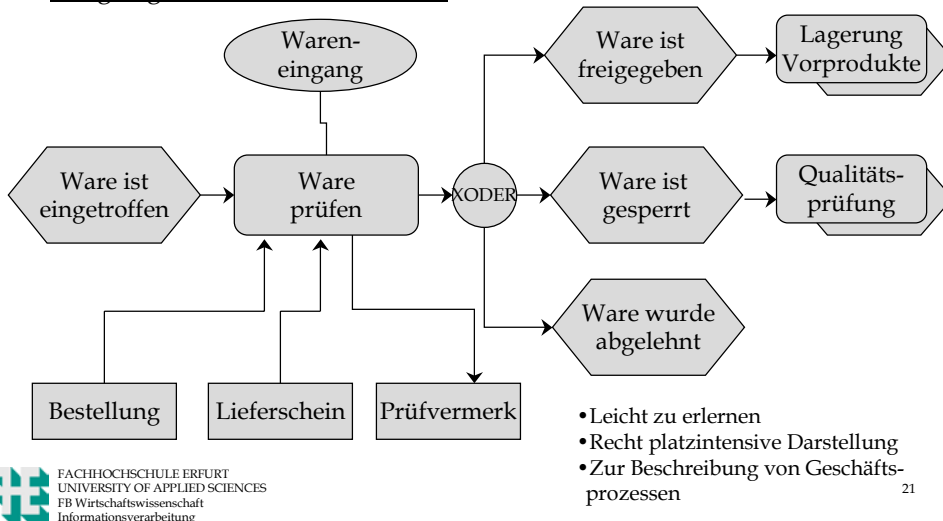
Ereignisgesteuerte Prozesskette I

Elemente der EPK



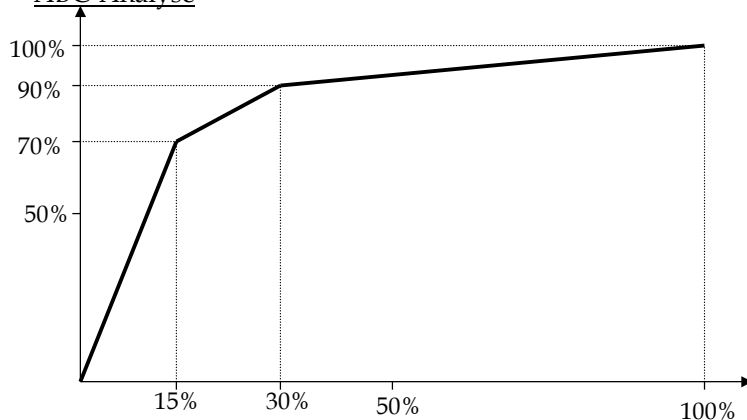
3. Entwicklung von Anwendungssoftware IST-Analyse/Techniken der Analyse

Ereignisgesteuerte Prozesskette II



3. Entwicklung von Anwendungssoftware IST-Analyse/Techniken der Analyse

ABC-Analyse



- Einfache Handhabung
- Hoher Aufwand zur Erhebung der Basisinformation
- Zur Trennung der wichtigen Vorfälle von den unwichtigen

3. Entwicklung von Anwendungssoftware SOLL-Konzept

Ziel:

Im SOLL-Konzept werden die Anforderungen an das geplante Anwendungssystem festgelegt (Pflichtenheft).

Fachentwurf (WAS soll das System leisten?)

- ABC-Analyse der erhobenen Geschäftsvorfälle
- Auf welche Weise lassen sich die aufgezeigten Mängel beseitigen?
- Welche wirtschaftlichen Vorteile sind vom neuen System zu erwarten?

IV-technischer Grobentwurf (WIE soll das System realisiert werden?)

- Realisierung (*Hardware, Kommunikationseinrichtungen, Programmiersprachen, Datenbanken, usw.*)
- Einführung (*Freigabe, Dokumentation, Schulung, Betreuung usw.*)
- Qualität (*Benutzerfreundlichkeit, Wartbarkeit, Zuverlässigkeit usw.*)
- Projektentwicklung (*Vorgehensmodell, Termine, Kosten, Auflagen usw.*)

3. Entwicklung von Anwendungssoftware SOLL-Konzept/Kreativitätstechniken

Die Aufstellung des Soll-Konzeptes erfordert u.a. Kreativität, die durch Kreativitätstechniken gefördert werden kann.

Regeln einer Brainstorming-Sitzung mit idealerweise 5 – 7 Teilnehmern:

- Keine Kritik an den vorgebrachten Ideen
- Kombinieren und Aufgreifen von bereits geäußerten Ideen
- Quantität ist erwünscht
- Freies Assoziieren und Phantasieren ist erlaubt

Intuitives und
assoziatives
Finden von
Lösungen

- Einfache Handhabung
- Geringe Kosten
- Gefahr der Abschweifung
- Für Zielformulierung und Lösungssuche.

3. Entwicklung von Anwendungssoftware SOLL-Konzept/Pflichtenheft

Das Pflichtenheft enthält alle betriebswirtschaftlichen und technischen Anforderungen an die einzusetzende Software.



Inhalt:

1. Portrait des Unternehmens
2. Ausgangssituation
3. Ziele aus Anwendersicht
4. Hardware-Anforderungen
5. Fachliche Anforderungen
6. Randbedingungen

3. Entwicklung von Anwendungssoftware Systementwurf/Programmspezifikation

Ziel:

Durch *Top-Down-Entwurf und Bottom-Up-Konkretisierung* soll ein strukturierter Entwurf des geplanten Systems erarbeitet werden, der als Vorlage für den konkreten Programmentwurf herangezogen wird.

Vorgehensweise

- Zerlegung der Gesamtfunktion in Module mit Teilfunktionen
- Jedes Modul wird in gleicher Weise weiter untergliedert
- Das Verfahren wird bis zu einer gewünschten Detaillierung fortgesetzt

Verfahren zur Darstellung von

- System- und Programmfunktionen
- Datenflüssen
- Kombinationen aus beidem
-

3. Entwicklung von Anwendungssoftware Systementwurf/Programmmentwurf

Ziel:

Erstellung einer detaillierten Vorgabe für die Programmierung.

Inhalte:

- **Datenorganisation**
Definition von Relationen, Primär-/Fremdschlüsseln ...
- **Eingabe**
Herkunft der Eingabedaten, Ablauforganisation der Datenerfassung, Festlegung von Formularen, Masken, Menüs
- **Verarbeitung**
Bestimmung der Betriebsarten, Nutzungsformen, Programmabläufe, ...
- **Ausgabe**
Definition von Bildschirmanzeigen, Drucklisten,



3. Entwicklung von Anwendungssoftware Programmmentwurf/Darstellungstechnik

Entscheidungs-Tabelle

Berechnung der Jahresprämie		R1	R2	R3	R4
B1	Alter ?	> 50	> 50	< 50	< 50
B2	Betriebszugehörigkeit ?	> 10	< 10	> 10	< 10
A1	Zahle Grundbetrag DM 2.000,-	X	X	X	X
A2	Zahle Treueprämie DM 1.000,-	X		X	
A3	Zahle Altersprämie DM 1.500,-	X	X		

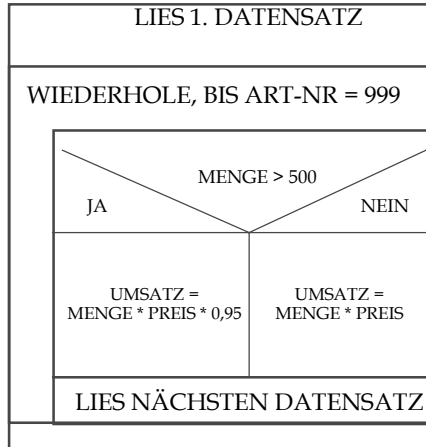
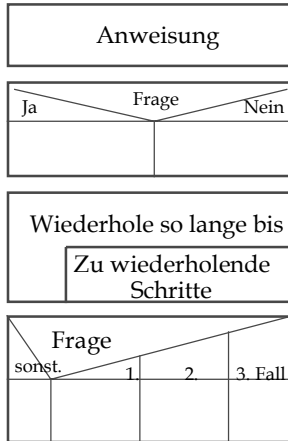
B = Bedingung; A = Aktion; R = Regel

Verändert übernommen aus: Horst Strunz: Entscheidungstabellentechnik, 1991



3. Entwicklung von Anwendungssoftware Programmwurf/Darstellungstechnik

Nassi-Shneiderman-Diagramme



3. Entwicklung von Anwendungssoftware Programmierung

Ziel:

Technische Entwicklung eines fachlich und syntaktisch fehlerfreien IV-Systems.

Inhalte:

- Codierung des Programms bzw. der Programmmodule
- Dokumentierung der Programme durch Einfügen von besonders gekennzeichneten Kommentaren.

```

ASSEMBLER          PRINT      NOGEN
PROGRAMM           START 0
                   DEVADDR=SYSRDR,
                   RECFORM=FIXUNB,
                   IOAREA1=CARDREC,...
                   DAVADDR=SYSLST,..
                   BALR      3,0          REGISTER 3 IS BASE
                   BEGIN                                           REGISTER
                   USING    *,3
                   OPEN     CARDFIL, REPFIL  OPEN FILES
                   MVC      PNRTREC, SPACES  MOVE SPACES TO
                                           OUTPUT RECORD
                   READLOOP .....
    
```

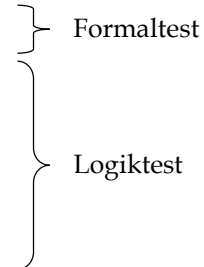
3. Entwicklung von Anwendungssoftware Systemtest

Ziel:
Eliminierung aller Programmfehler in der erstellten Software



Testarten:

- Programmtest
- Einzeltest (Modultest)
- Integrationstest (Komponententest)
- Systemtest
- Abnahmetest



Systemeinführung

Ziel:
Reibungslose Inbetriebnahme des neuen Systems und die Ablösung von Altsystemen.

Die Einführung kann erfolgen:

- mit dem gesamten Anwendungssystem einschließlich aller Daten zu einem Stichtag und gleichzeitiger Beendigung des Altsystembetriebs (Big Bang),
- stufenweise mit nur einem Teil des Anwendungssystems, nur einem Teil der Daten oder nur einem Teil der Nutzer,
- als Parallellauf unter gleichzeitiger, zeitlich begrenzter Fortführung des Altsystems.

