

Beispiele: Fortran-Sourcecode (I)

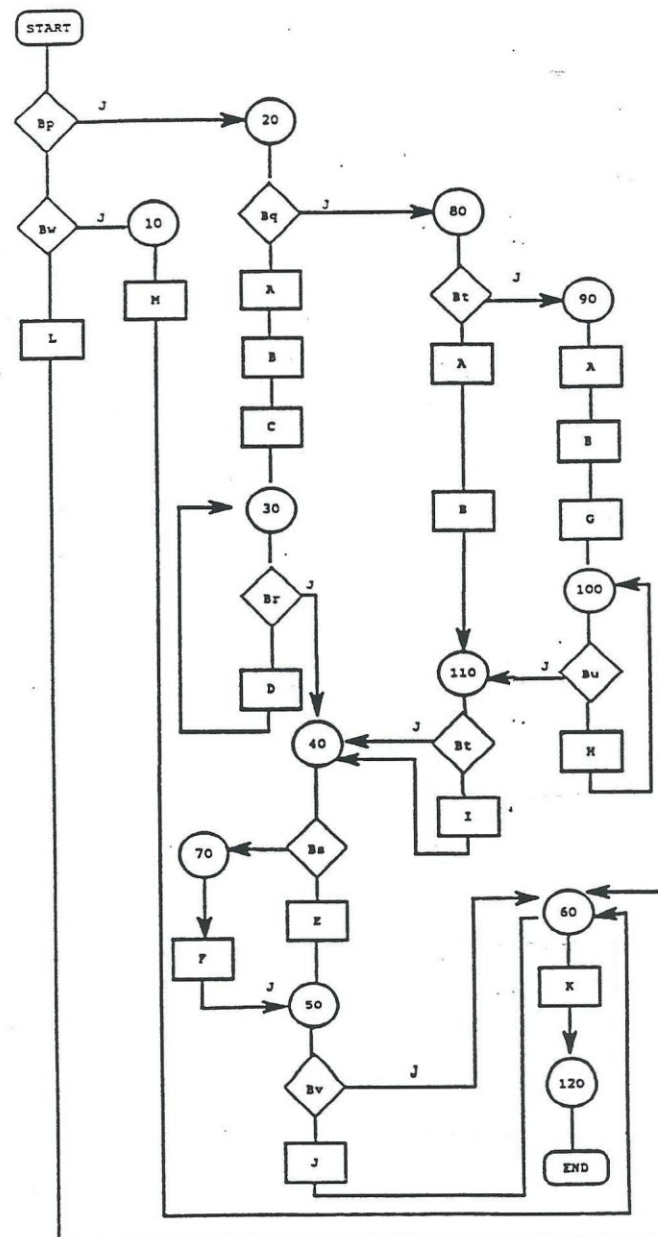
unstrukturiert	maschinell restrukturiert (ohne Optionen)
<pre> program test1 c implicit integer (a-z) c if (p) goto 20 if (w) goto 10 L = L + 1 goto 60 10 M = M + 1 goto 60 20 if (q) goto 80 A = A + 1 B = B + 1 C = C + 1 30 if (.not.r) goto 40 D = D + 1 goto 30 40 if (s) goto 70 E = E + 1 50 if (.not.v) goto 60 J = J + 1 60 K = K + 1 goto 120 70 F = F + 1 goto 50 80 if (t) goto 90 A = A + 1 B = B + 1 goto 110 90 A = A + 1 B = B + 1 G = G + 1 100 if (.not.u) goto 110 H = H + 1 goto 100 110 if (.not.t) goto 40 I = I + 1 goto 40 120 end </pre>	<pre> C C INOFOR 4.1 program test2 C c implicit integer (a-z) C c if (.NOT.p) then if (.NOT.w) then L = L + 1 GOTO 60 endif M = M + 1 else IF (q) GOTO 80 A = A + 1 B = B + 1 C = C + 1 C INNOVATOR WHILE 1 if (.not.(.NOT.r)) goto 2 D = D + 1 goto 1 2 continue C END INNOVATOR WHILE 40 CONTINUE IF (s) GOTO 70 E = E + 1 50 CONTINUE if (.NOT.v) then J = J + 1 endif endif 60 CONTINUE K = K + 1 GOTO 120 70 CONTINUE F = F + 1 GOTO 50 80 CONTINUE if (.NOT.t) then A = A + 1 B = B + 1 else A = A + 1 B = B + 1 G = G + 1 C INNOVATOR WHILE 3 if (.not.(.NOT.u)) goto 4 H = H + 1 goto 3 4 continue C END INNOVATOR WHILE endif IF (t) GOTO 40 I = I + 1 GOTO 40 120 CONTINUE stop end </pre>

Beispiele:

Fortran-Sourcecode (II)

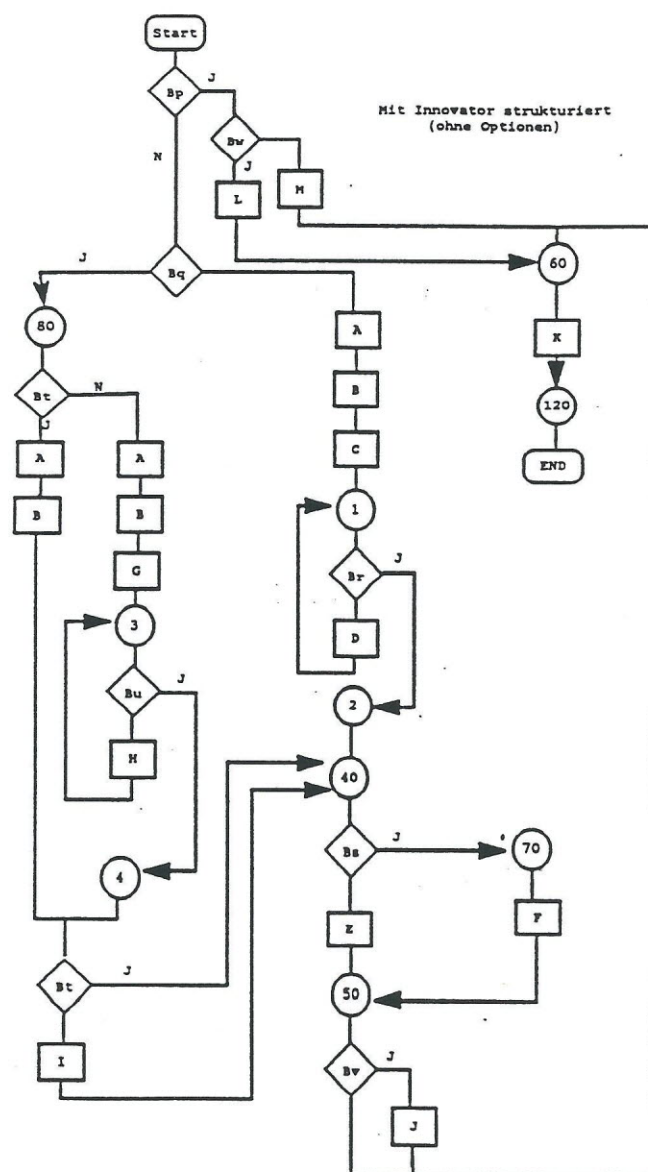
maschinell restrukturiert (alle Optionen)		manuell restrukturiert
C C INOPOR 4.1	:	program test6
	:	C
program test3	:	implicit integer (a-z)
C C	:	C
C C INOPOR: New logical Variables:	:	
C C	:	if (p) then
LOGICAL GSOD	:	a = a +1
LOGICAL GSOD	:	b = b +1
C C	:	if (q) then
implicit integer (a-z)	:	if (t) then
GSOD = .FALSE.	:	g = g +1
C C	:	20 if (u) then
if (.NOT.p) then	:	h = h +1
if (.NOT.w) then	:	goto 20
L = L +1	:	endif
K = K +1	:	i = i +1
STOP	:	endif
endif	:	else
M = M +1	:	c = c +1
else	:	30 if (r) then
GSOD = .FALSE.	:	d = d +1
GSOD = q	:	goto 30
if (.NOT.GSOD) then	:	endif
A = A +1	:	endif
B = B +1	:	if (s) then
C = C +1	:	f = f +1
C INNOVATOR WHILE	:	else
1 if (.not.(.NOT.r)) goto 2	:	e = e +1
D = D +1	:	endif
goto 1	:	if (v) then
continue	:	j = j +1
C END INNOVATOR WHILE	:	endif
40 CONTINUE	:	else
if (s) then	:	if (w) then
F = F +1	:	m = m +1
GSOD = .TRUE.	:	else
endif	:	l = l +1
if (.NOT.GSOD) then	:	endif
E = E +1	:	endif
endif	:	k = k +1
if (.NOT.v) then	:	end
J = J +1	:	
endif	:	
endif	:	
endif	:	
if (.NOT.GSOD) then	:	(--> Fortsetzung von links)
K = K +1	:	C INNOVATOR WHILE
STOP	:	3 if (.not.(.NOT.u)) goto 4
endif	:	M = M +1
GSOD = .FALSE.	:	goto 3
if (.NOT.t) then	:	4 continue
A = A +1	:	C END INNOVATOR WHILE
B = B +1	:	endif
else	:	IF (t) GOTO 40
A = A +1	:	I = I +1
B = B +1	:	GOTO 40
G = G +1	:	stop
(Fortsetzung -->)	:	end

Beispiel: PAP (Programmablaufplan) (unstrukturiert)



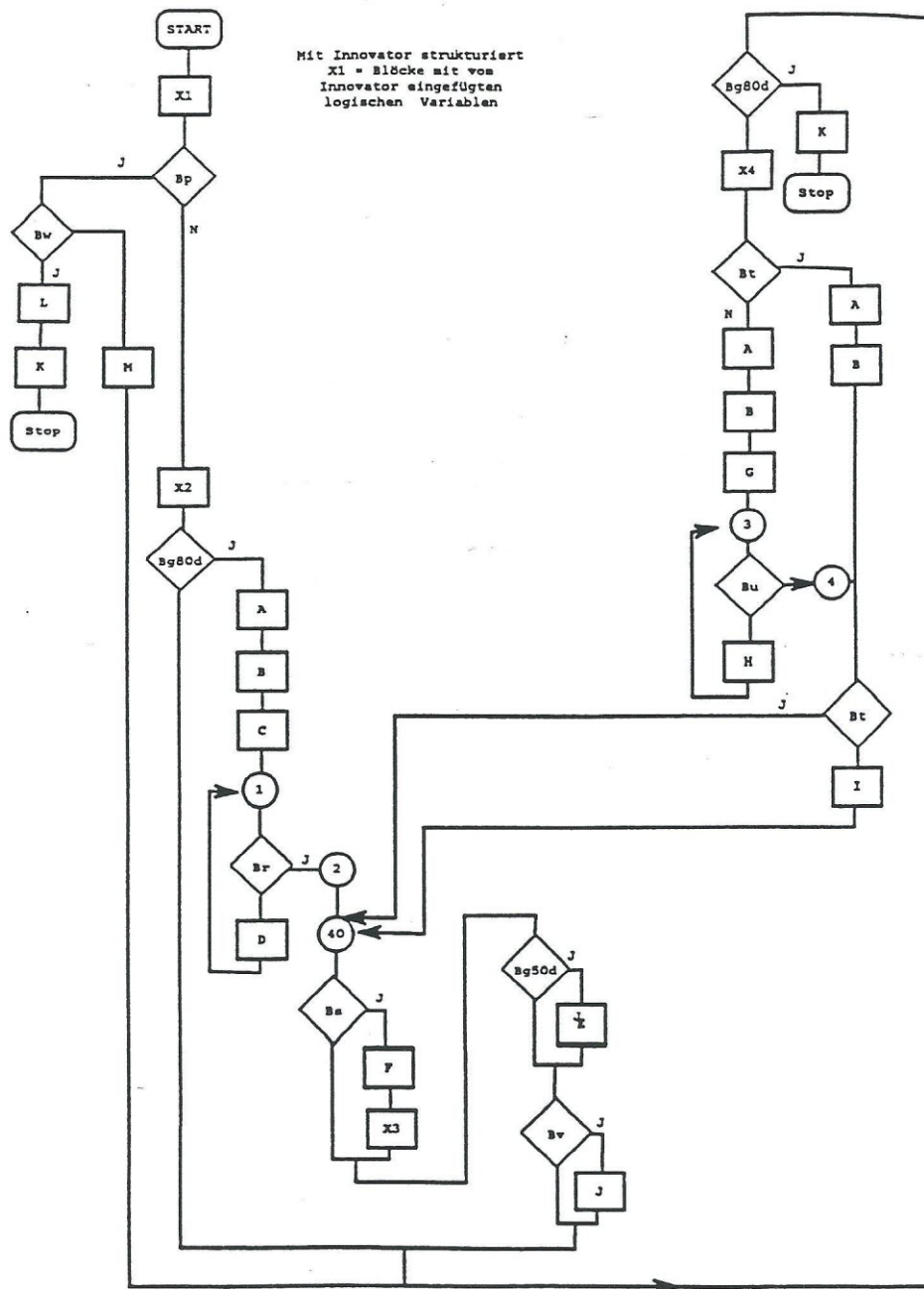
Beispiel: PAP (Programmablaufplan)

(maschinell restrukturiert,
Stufe 1)

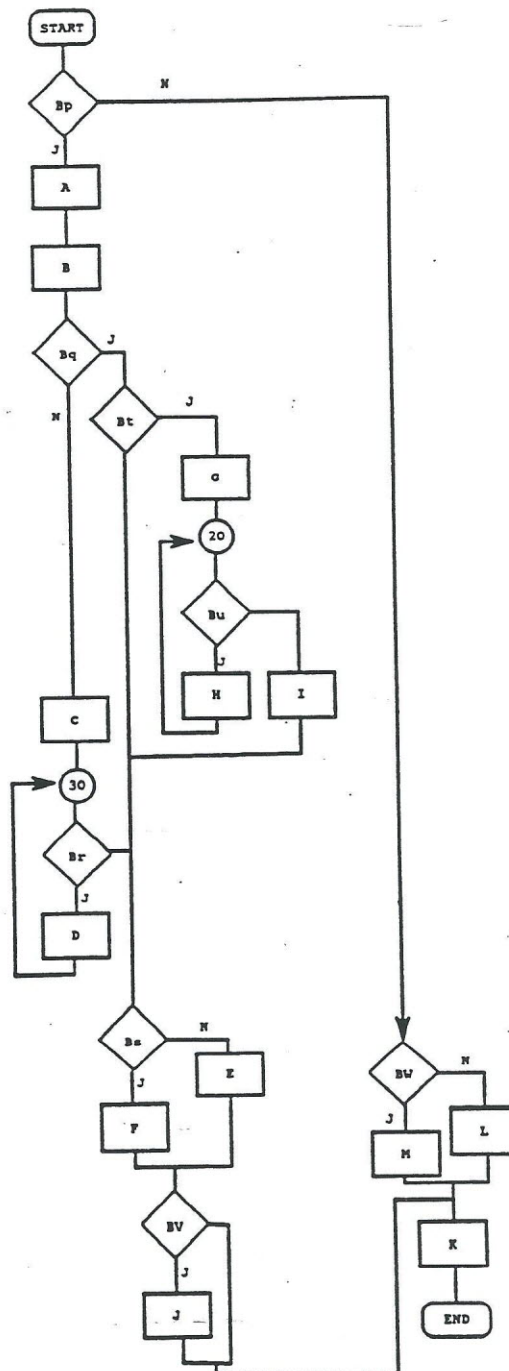


Beispiel: PAP (Programmablaufplan)

(maschinell restrukturiert,
Stufe 2)



Beispiel: PAP (Programmablaufplan) (manuell restrukturiert)



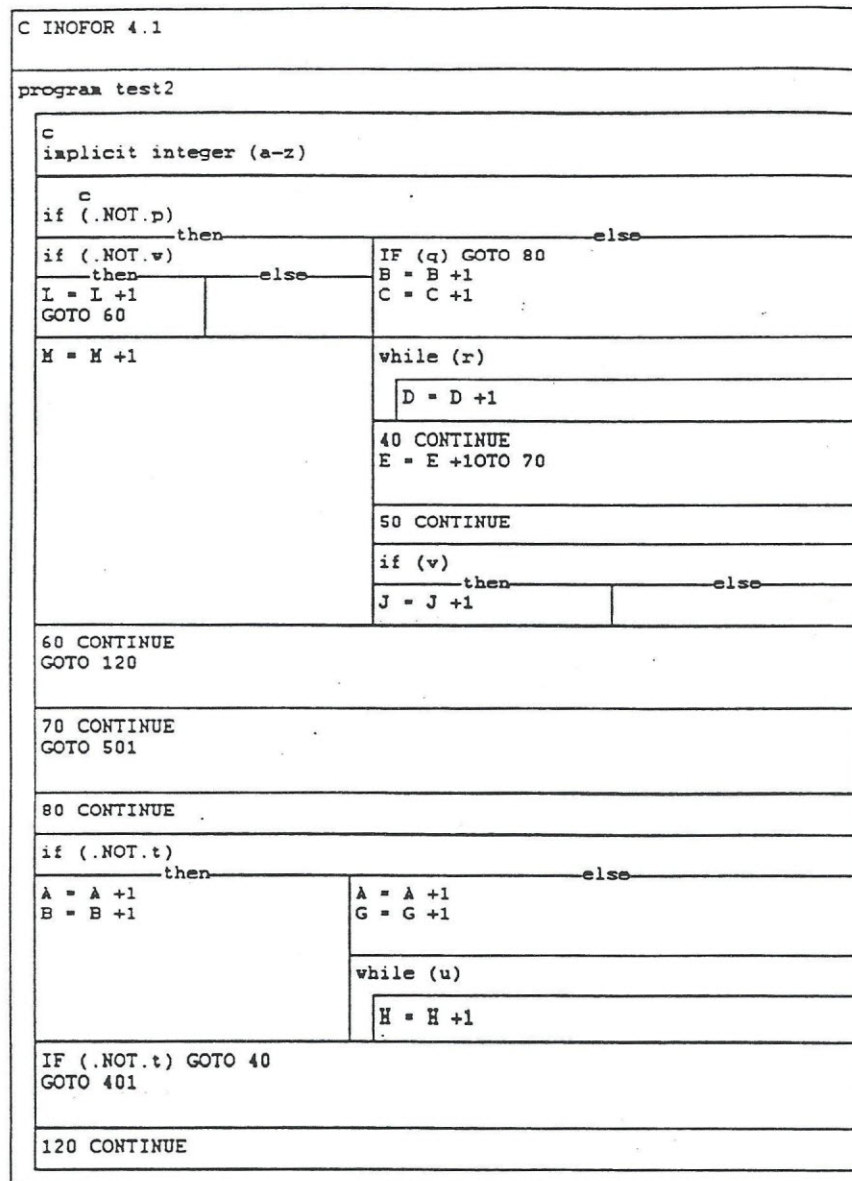
Beispiel: (Pseudo-)NSD (Struktogramm)

(maschinell restrukturiert,
Stufe 1)

Innovator V4.1

MID GmbH

Tue May 26 15:14:10 1992



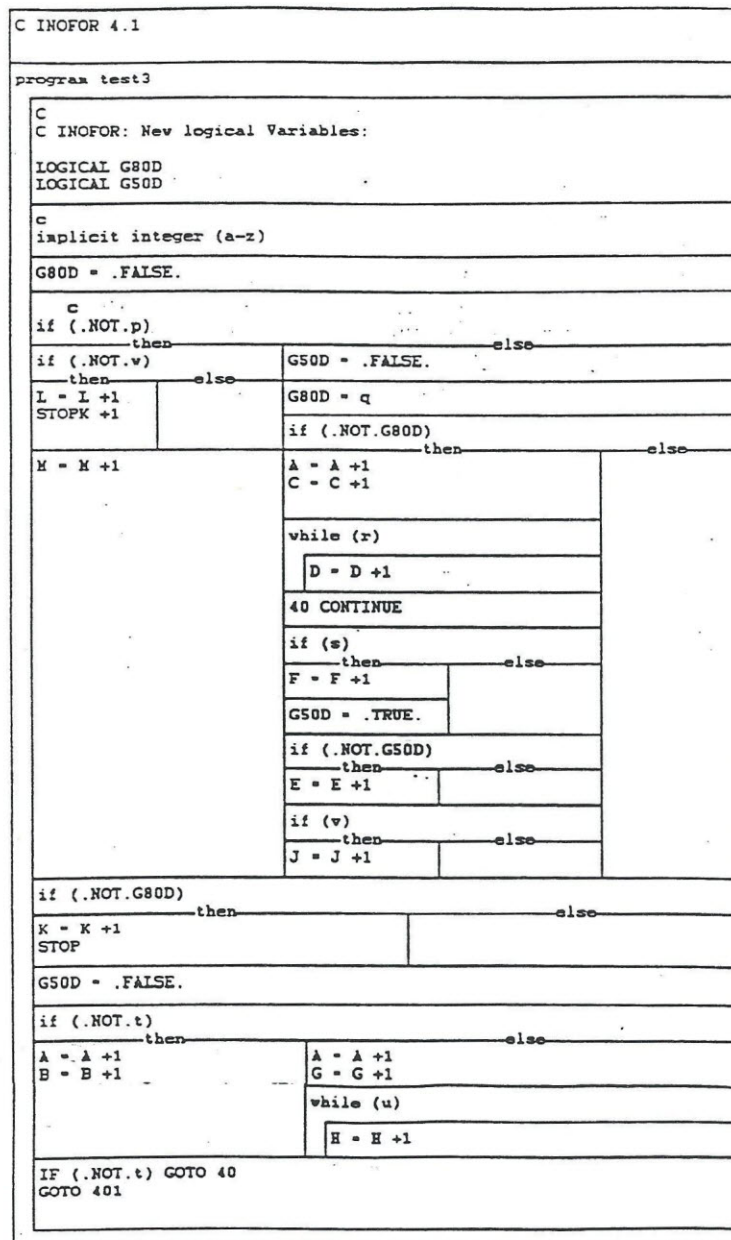
Beispiel: NSD (Struktogramm)

(maschinell restrukturiert,
Stufe 2)

Innovator V4.1

MID GabH

Thu Jun 04 18:12:47 1992

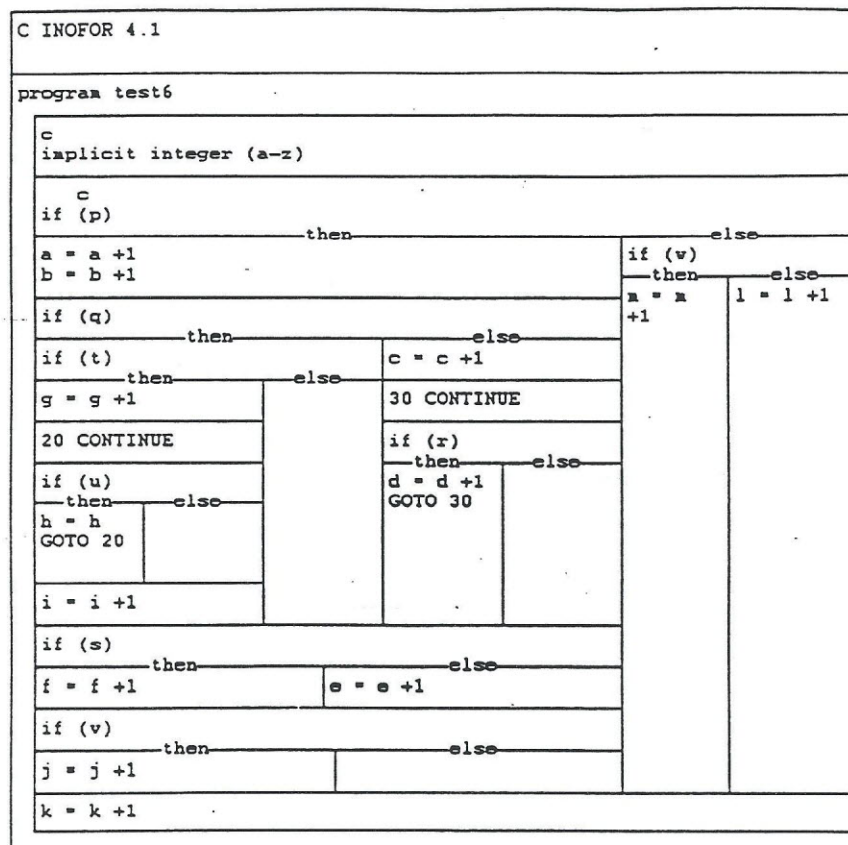


Beispiel: NSD (Struktogramm) (manuell restrukturiert)

Innovator V4.1

MID GmbH

Tue May 26 15:21:16 1992



Statische Analyse (I)

Beispielprogramme	TEST1	TEST2	TEST3	TEST6
Anzahl Labels angesprungen	17	11	6	2
Anzahl Zeilen insgesamt	39	60	71	47
Kommentarzeilen	2	8	11	2
LOGICAL	0	0	2	0
CONTINUE	0	8	3	0
STOP	0	1	3	0
GOTO	17	11	6	2
Logical IF	9	5	3	0
Block IF	0	4	8	8
ELSE	0	2	2	4
ENDIF	0	4	8	8
Assignment	17	16	22	13
Anzahl executable statements	44	52	56	39

Statische Analyse (II)

Programm PM6SZE	unstrukt.	strukt. (ohne Op.)	strukt. (mit Op.)
Anzahl Labels angesprungen	213	153	152
Anzahl Zeilen insgesamt	2001	2138	3913
Kommentarzeilen	966	1029	1553
LOGICAL	2	2	29
CONTINUE	133	98	115
STOP	2	2	6
GOTO	189	129	108
Logical IF	141	72	65
Block IF	76	141	369
ELSE	23	32	87
ENDIF	76	141	369
Assignment	305	305	853
Anzahl executable statements	1029	1017	2128

Restrukturierungsregeln (I)

- **Elimination nicht verwendeter Labels**
- **Elimination dynamischer Steuerungselemente**
- **Untersuchung jeder Rückwärts-Goto-Anweisung**
- **Untersuchung jeder Vorwärts-Goto-Anweisung**
- **Beseitigung von Strukturbrüchen**
- **Vereinfachen und Zusammenfassen**

Restrukturierungsregeln (II)

- **Entfernung toter Zweige**
- **Entfernung unbenutzter Datenfelder und -strukturen**
- **Explizite Definition aller verwendeten Datenfelder**
- **Überarbeitung bestehender und Hinzufügung neuer Kommentarblöcke**

Elimination dynam. Steuerungselemente

(Bsp. Arithmetisches IF)

unstrukturierter Sourcecode

```
2   Block 2
   ...
   IF (n) 1,2,3
   ...
3   Block 3
   ...
1   Block 1
   ...
```

wird restrukturiert in

```
IF (n .LT. 0) THEN
  BLOCK 1
ELSE IF (n. EQ. 0) THEN
  BLOCK 2
ELSE
  BLOCK 3
ENDIF
```

Elimination dynam. Steuerungselemente

(Bsp.: Arithmetisches IF)

unstrukturiert

```
      if(effd(j)-effd(i)) 4,5,6
4      l=i
      k=j
      goto 7
5      if(endd(j)-endd(i)) 4,4,6
6      l=j
      k=i
7      continue
```

maschinell
restrukturiert

```
      g4d = .false.
      if (effd(j)-effd(i).gt.0) then
          l=j
          k=i
      else
          if (effd(j)-effd(i).ne.0) then
              l=i
              k=j
          else
              if (endd(j)-endd(i).le.0) then
                  g4d = .true.
                  l=i
                  k=j
              endif
              if (.not.g4d) then
                  l=j
                  k=i
              endif
          endif
      endif
      endif
```

manuell
restrukturiert

```
      IF ((effd(j) - effd(i)) .LT. 0) THEN
          l = i
          k = j
      ELSEIF ((effd(j) - effd(i)) .EQ. 0) THEN
          IF (((endd(j) - endd(i)) .LE. 0) THEN
              l = i
              k = j
          ELSE
              l = j
              k = i
          ENDIF
      ELSE
          l = j
          k = i
      ENDIF
```

Elimination dynam. Steuerungselemente

(Bsp.: Computed GoTo)

unstrukturierter Sourcecode

```
10  Block A
...
30  Block C
...
    GOTO (10,20,30,40)
...
40  Block D
20  Block B
...
```

wird restrukturiert in

```
IF (s.EQ.1) THEN
    Block A
ELSE IF (s.EQ.2) THEN
    Block B
ELSE IF (s.EQ.3) THEN
    Block C
ELSE IF (s.EQ.4) THEN
    Block D
    Block B
ENDIF
```

Elimination dynam. Steuerungselemente

(Bsp.: Assigned GoTo)

unstrukturierter Sourcecode

```
10  Block A
    ...
    ASSIGN 10 TO x
    IF (s.EQ.0) ASSIGN 20 TO x
    ...
    GOTO x (10,20)
    ...
20  Block B
    ...
```

wird restrukturiert in

```
x = 10
IF (s.EQ.0) THEN
    x = 20
ENDIF
...
IF (x.EQ.10) THEN
    Block A
ELSE IF (x.EQ.20) THEN
    Block B
ENDIF
```

Vereinfachung von Schleifen

strukturierte aber komplizierte Schleife

```
...
counter = 1
error = .FALSE.
WHILE ((.NOT. error) .AND.
      (counter .LE. max)) DO
...
  Block A
...
  IF (Fehlerbedingung) THEN
    error = .TRUE.
  ENDIF
  IF (.not. error) THEN
...
    Block B
    counter = counter + 1
  ENDIF
ENDWHILE
erster Befehl nach Ende
der WHILE-Schleife
...
```

vereinfachte strukturierte Schleife

```
...
counter = 1
DO 10 counter = 1, max
...
  Block A
...
  IF (Fehlerbedingung) GOTO 20
...
  Block B
10 CONTINUE
20 CONTINUE
erster Befehl nach BREAK (hier: GOTO 20)
oder Ende der DO-Schleife
...
```

Zusammenfassung von IF-Statements

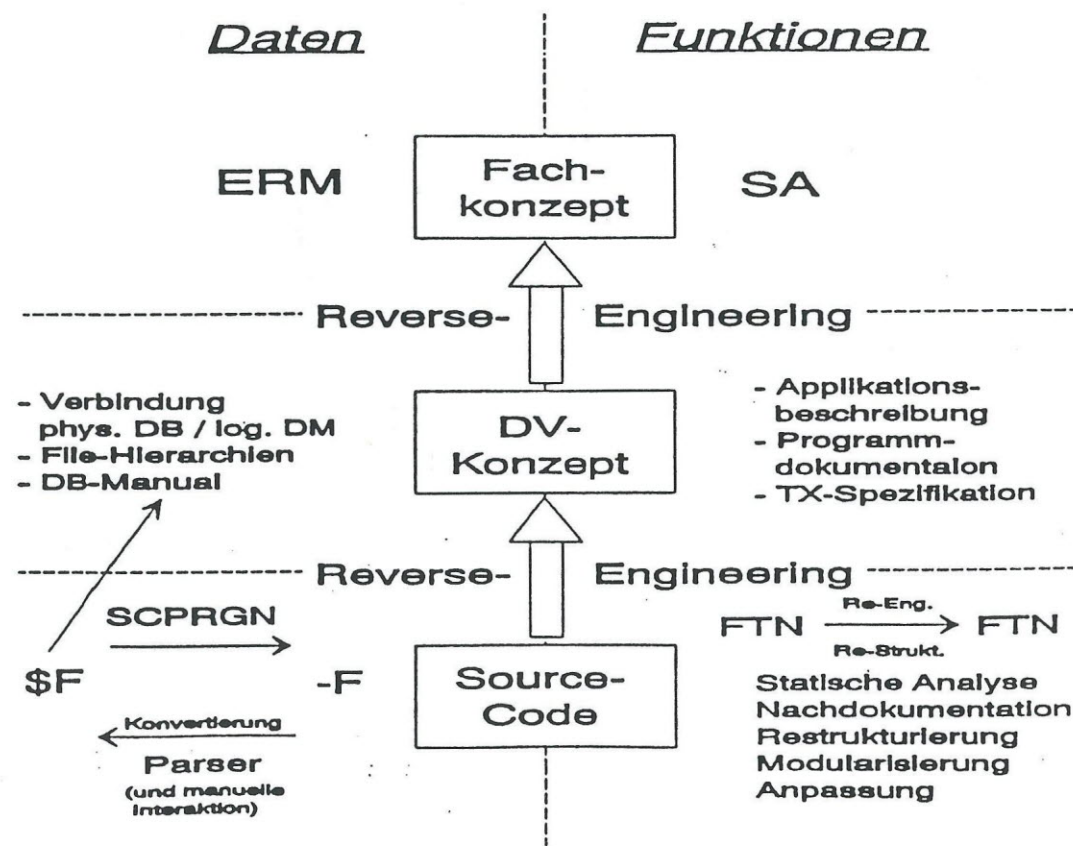
Beispiel aus Originalprogramm:

```
if (effd(j).gt.endd(i)) goto 2
if (effd(i).gt.endd(j)) goto 2
if (or(days(i),days(j)).eq.127) goto 2
```

wird restrukturiert in

```
if ((effd(j) .gt. endd(i)) .or.
*   (effd(i) .gt. endd(j)) .or.
*   (days(i) .eq. 127) .or.
*   (days(j) .eq. 127)) goto 2
```

Ebenen der Softwaresanierung für die UNISYS-Passagesysteme der Lufthansa

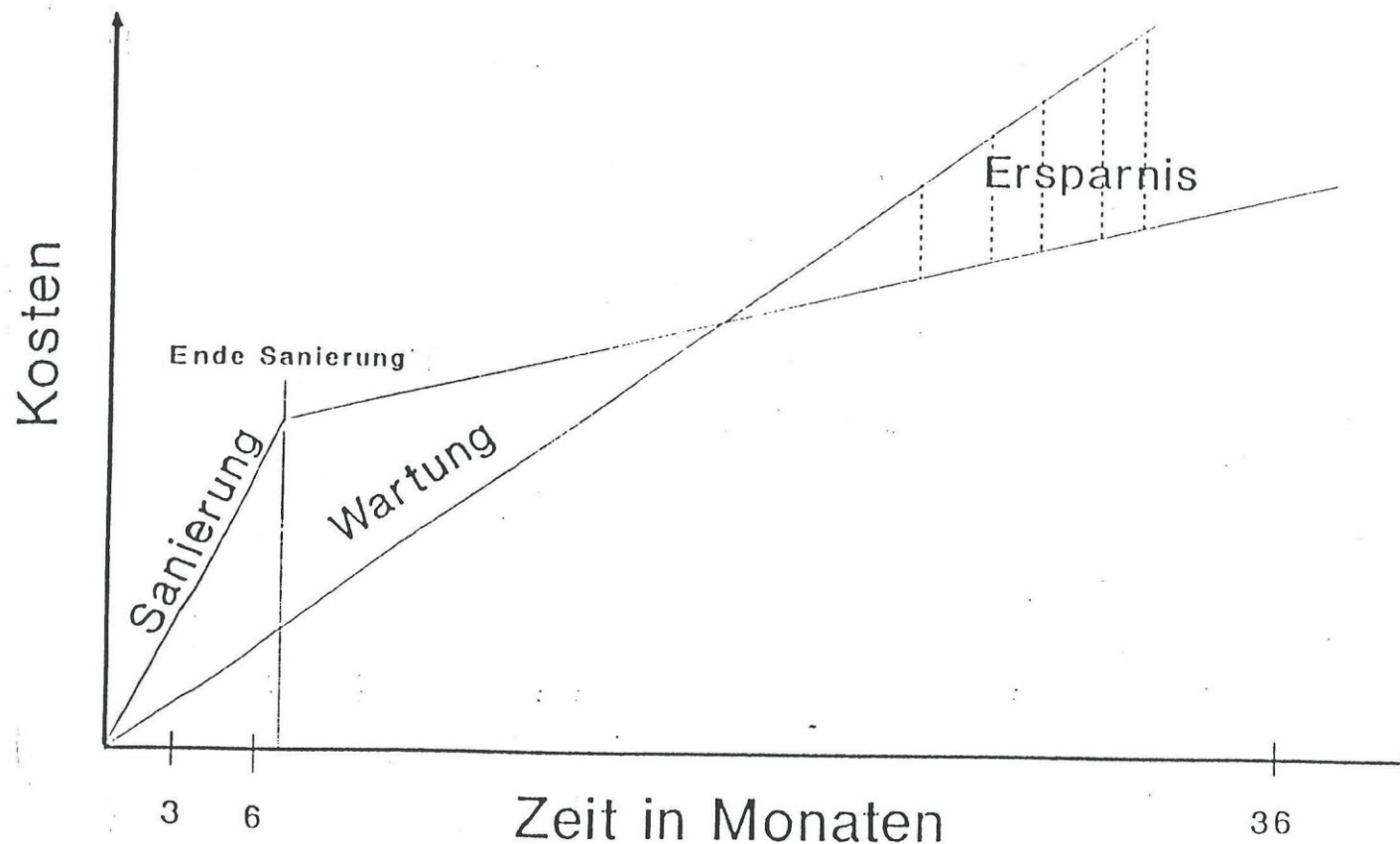


Wirtschaftlichkeitsbetrachtungen

Es gibt kein Standardverfahren zur Schätzung des Sanierungsaufwandes !

- **LH*SYS: 30 BJ bei 400 Elementen**
- **LH*RES: hat 2000 Elemente**
- **LH*CKI**
- **...**

Sanierungsgewinn (idealisiert)



Einflußfaktoren auf die Wirtschaftlichkeit

- **Programmgröße (loc, Anzahl der Programme)**
- **Anzahl I/O-Operationen**
- **Komplexität (Algorithmen/Datenstrukturen)**
- **Anzahl der Schnittstellen (viel/wenig, breit/schmal)**

Einflußfaktoren auf die Wirtschaftlichkeit

- . Modularität (viel/wenig, groß/klein)**
- . Programmart (Realtime, Online, Batch)**
- . Grad der Automatisierung**
- . Erfahrung des Sanierungspersonals**
- . Programmiersprache**

Grenzen der Sanierung

- **nicht alles sanierbar**
- **Kosten/Nutzen**
- **Grenze Produktivitätssteigerung $\approx 30\%$**
- **kein automatisierter Vorgang**
- **Zielcode z.T. schlechter wartbar**
- **Entfremdung**
- **Programmgröße steigt**

Weiteres Vorgehen

- **Strategische Informationsplanung**
- **Sanierungsentscheidung**
- **Festlegung der Sanierungsstrategie**
- **Festlegung der Sanierungsorganisation**
- **Interdependenzen klären**
- **QS-Analyse durchführen**
- **Toolauswahl**