

## Das Beste aus 30 Jahren Excel-Entwicklung (Teil 1 von 2)

Aus der Vielzahl umgesetzter Lösungen sind 2 Modelle von besonderer Bedeutung: Automatisierung der Zeitfortschreibung (1) und Kundenerfolgsrechnung (2).

### (1) Automatisierung der Zeitfortschreibung in der Statistik

Produzierende Unternehmen benötigen aus mehreren Gründen eine ordentliche Statistik. Excel ist für eine rationelle Lösung besonders geeignet, weil die Rechenarbeit, anders als in herkömmlichen Lösungen, nicht mehr programmiert werden muss. Denn Excel ist bereits ein fertiges Rechenprogramm. Außerdem enthält es ein integriertes Automatisierungswerkzeug, mit dem die Bedienung und alle Berechnungen und Auswertungen automatisierbar sind und so die Vision „PC einschalten, klicken, erfassen, fertig“ verwirklicht wird. Lassen Sie Excel arbeiten, anstatt mit Excel zu arbeiten. Excel ist weit verbreitet, allerdings fällt auf, dass in den meisten Lösungen viel Zeit für die Erstellung, Bearbeitung und Suche von Tabellen und Dateien verloren geht. Leider kommen die meisten Eigenentwickler nicht über die Makrohürde hinweg, zur Freude aller Geschäftemacher, die sagen „mit Excel geht das nicht“, und viele Unternehmen zahlen täglich(!) mehrere Tausend Euro Lizenzgebühren. Dabei könnte auch in größeren Unternehmen der gesamte Statistik- und Kalkulationsbereich gut mit Excel abgedeckt werden.

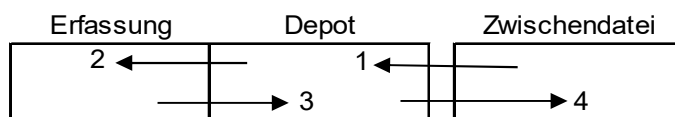
Referenz: Zeitaufwand auf  $\frac{1}{4}$  reduziert.

Aufgabe: Erfassung von Tagesdaten, Auswertung und zeitliche Fortschreibung automatisiert.

Beispiele: Tages-Produktionsmengen; Prüfung von Frachtrechnungen und ähnliche Aufgaben.

Problem: Vorhandene Excel-Statistik-Lösungen sind entweder **zeitorientiert** (Monatsdateien mit Produktblättern) oder **produktorientiert** (Produktdateien mit Monatsblättern) strukturiert. So kommen mehrere hundert Tabellen zusammen. Die Pflege der vielen Dateien und Tabellen, voller Formeln und Verknüpfungen, nehmen mit Monats- und Jahresstatistiken viel Zeit in Anspruch. Die eigentliche Datenerfassung ist dem gegenüber nahezu unerheblich.

Lösung: **Datenerfassung** in vorgeschalteter Tabelle, gültig für alle Zeiträume, **ohne Zeitkennung** im Dateinamen. Nach der Erfassung werden die relevanten Daten makrogesteuert in eine **Zwischendatei mit Monats- und Jahreskennung im Dateinamen** gespeichert, die Erfassung geleert. Beim nächsten Aufruf wird der Datenstand aus der Zwischendatei geholt:



Datenfluss: Beim Aufruf der Erfassungsdatei wird die Zwischendatei geöffnet, der Datenstand des eingestellten Monats in das Depotblatt kopiert (1) und die Zwischendatei ohne Speichern geschlossen. Beim Anwählen der Erfassungstabelle wird der Datenstand des gewählten Tages aus dem Depot in die Erfassungstabelle kopiert (2). Nach beendeter Erfassung wird der neue Datenstand in das Depotblatt kopiert (3), die Erfassungsspalte(n) geleert. Beim Verlassen der Erfassungsdatei wird der Depot-Datenstand als Zwischendatei im Überschreibmodus gespeichert (4), das Depot geleert.

Zeiteinstellung (Tag, Monat, Jahr), Datenfluss und Bedienung erfolgen über individuell angelegte Schaltflächen, denen VBA-Makros zugewiesen sind. Die Quelldaten liegen entweder digital oder auf Papier vor. Nachstehendes Beispiel zeigt die Papierform. Diese Methode ist nach wie vor verbreitet dient als elementare Datenerfassung als bester Archivnachweis. Bei digitaler Quelldaten-Verfügbarkeit sind die Monats- und Jahresstatistiken hoffentlich ebenfalls verfügbar. Ansonsten ist ebenfalls eine Weiterverarbeitung der Daten in Excel möglich.

Beispiel: Produktionsstatistik (Praxisbeispiel, hier auf die Produktionsmengen reduziert).  
Das Modell besteht aus 4 Dateien: *Start, Erfassung, Monat, Jahr*.

Schritt 1 von 4: Startdatei mit 2 Blättern und folgenden Elementen:

**Blatt Wahl:**

<b>Firmenname</b>	Monat Ab	Monat Auf	Ende
<b>Produktion</b>	August 2019		
<b>Erfassung</b>			
<b>Monatsstatistik</b>			
<b>Jahresstatistik</b>			

**Blatt Stamm:**

	C	G	H
11	Firmenname	11	1 Januar
12	August	12	2 Februar
13	8		usw. ...bis...
14	2019	22	12 Dezember
21	C:\Prod		
31	C:\Prod\Start.xlsm		
32	C:\Prod\Erfassung.xlsm		
33	C:\Prod\Monat.xlsm		
34	C:\Prod\Jahr.xlsm		
41	C:\Prod\zProd-8-2019.xlsx		

Blatt *Wahl*: Auf den farbig markierten Zellen liegen Schaltflächen mit gleichnamiger Makrozuweisung. Firmenname, Monat, Jahr sind auf das Blatt *Stamm* verknüpft.

Blatt *Stamm*: C11, C13, C14 sind einmalig einzugeben.

C12 wird zu C13 aus G11:H22 per `=SVERWEIS(C13;G11:H22;2;)` gezogen.

C13, ggf. C14 werden per Auf-/Ab-Makro geändert: `ActiveCell.Value = ActiveCell.Value + 1`

C31-C34 werden wie folgt hinterlegt: C31: `=C21&"\Start.xlsm"` usw.,

C41: `=C21&"\zProd-"&C13&"-"&C14&".xlsx"`.

Falls eine Erfassung von Tages-Endbeständen erfolgt, ist folgende Erweiterung erforderlich;

C15: `=WENN(C13=1;12;C13-1)` C16: `=WENN(C13=1;C14-1;C14)`

C42: `=C21&"\zProd-"&C15&"-"&C16&".xlsx"`.

Der Stammbereich C11:C110 wird bei jedem Datei-Öffnen (Erfassung, Monat, Jahr) makrogesteuert in die Zieldatei, Blatt *Stamm*, umkopiert (Einfügen-Werte).

Per Auto\_Open-Makro wird eingesetzt:

```
ChDir (ThisWorkbook.Path)
Application.Calculation = xlAutomatic
Range("C21").Select
ActiveCell.Formula = ActiveWorkbook.Path
```

Für die Startdatei wird in der Windows-Oberfläche ein Symbol angelegt, bei dessen Anklicken Excel aktiviert und die Startdatei geöffnet wird.

Die Makros zum Aufruf der Erfassungsdatei, Monats- und Jahresdatei enthalten jeweils ein Top-Makro, d. h. dieses in der Folgedatei stehende Makro ist in den Öffnen-Vorgang eingebaut:

`Workbooks.Open Filename:=Cells(32, 3)`

`Application.Run "Erfassung.xlsm"!TopErfassung"`

Außerdem sollte sichergestellt sein, dass die Arbeitsdateien nicht gesucht werden, sondern per Schaltfläche geöffnet werden. Dazu dient jeweils ein Auto\_Open-Makro (Beispiel):

```
Sub Auto_Open()
    msgbox "Bitte die Eingabedatei aus der Startdatei heraus aufrufen"
    Sheets("Stamm").Select
    Workbooks.Open Filename:=Cells(31, 3)
    Application.Windows(2).Close (False)
End Sub
```

## Schritt 2 von 4: Erfassungsdatei mit 6 Blättern und folgenden Elementen:

**Blatt Wahl**

Firmenname										<b>Zurück zur Startdatei</b>																				
Produktion										August 2019																				
1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.	17.	18.	19.	20.	21.	22.	23.	24.	25.	26.	27.	28.	29.	30.	31.

**Blatt Wahl 2**

Firmenname										<b>1. August 2019</b>									
Produktion																			
Produktion																			

**Blatt Prod**

	C	D	E	F	G	H	S	AW	AX	BS	CW	CX
2	Firmenname						<b>Zurück</b>					
7	Produktion						<b>1. August 2019</b>					
9			Menge		kg je	Menge	Einheiten:			Kilo:		
10	Nr.	Produkt	Einh.	Einheit	Einh.	kg	1	31	Ges.	1	31	Ges.
11	110	Produkt 110, 250g	<b>1.280</b>	Steige	5,00	6.400	1280		1280	6400		6400
12												

= Hinterlegung   
  = Eingabe   
  = Berechnung oder zur Information

Die Erfassung erfolgt entweder direkt in der Produktion oder alternativ über dort erstellte Tagesberichte in Papierform. Weitere Blätter: *Depot*, *Formel*, *Format*.

Im Blatt *Prod* sind die Zellen E7-H7 auf das Blatt *Wahl2* verknüpft.

Im Blatt *Stamm* werden in C11-C110 die Stammdaten aus der Startdatei einkopiert.

Im Blatt *Depot* werden die Erfassungsdaten deponiert.

Im Blatt *Formel* werden die Formeln zu den Spalten H, AX, CX hinterlegt.

Im Blatt *Format* wird das Zeilenformat C-CX hinterlegt.

Beim Anklicken einer der 31 Tages-Schaltflächen wird die Tageszahl im Blatt *Wahl2* makrogesteuert eingetragen (Monat und Jahr sind auf das Blatt *Stamm* verknüpft).

Beim Anklicken der Schaltfläche *Produktion* sind folgende Makropassagen erforderlich:

- Einkopieren des letzten Datenstandes aus dem Blatt *Depot* in das Blatt *Prod*
- Einsetzen der Daten des gewählten Tages aus S-AW in die Spalte E
- Einsetzen der Zeilenformate (11-1010 = 1000 Zeilen)
- Einsetzen der Kilo-Formel in H11-H1010

Beim Aufruf der Erfassungsdatei aus der Startdatei heraus wird das Top-Makro durchgeführt:

Sub TopErfassung()

Sheets("Stamm").Select

strPfad = Cells(41, 3)

If Dir\$(strPfad, vbNormal) <> "" Then

Workbooks.Open Filename:=Cells(41, 3)

‘ Daten A1-CW1010 in das Blatt *Depot* kopieren, Zwischendatei schließen

End If

End Sub

Makroabschnitt, um bereits erfasste Daten in die Eingabespalte zu setzen:

```
Sub uProdInTabSetzen()  
  Sheets("Prod").Select  
  s = 19  
  While s < 50  
    If Cells(10, s) = Cells(7, 5) Then  
      Range(Cells(11, s), Cells(1010, s)).Select  
      ' kopieren, in E11 einfügen (Werte)  
      Exit Sub  
    End If  
    s = s + 1  
  Wend  
End Sub
```

Makroabschnitt, um bei *Zurück* die Erfassung in die Monatstabellen zu setzen:

```
Sub uErfassungSichern()  
  Sheets("Prod").Select  
  s = 19  
  While s < 50  
    If Cells(10, s) = Cells(7, 5) Then  
      Range("E11:E1010").Select  
      Selection.Copy  
      Cells(11, s).Select  
      ' einfügen (Werte)  
      Range("H11:H1010").Select  
      Selection.Copy  
      Cells(11, s + 52).Select  
      ' einfügen (Werte)  
      Exit Sub  
    End If  
    s = s + 1  
  Wend  
End Sub
```

Anschließend wird bei *Zurück* C11-CX1010 in das Blatt *Depot* C11 kopiert und die Erfassungstabelle C11-CW1010 gelöscht: *Selection.Clear*.

Die Erfassungstabelle darf zu beliebigen Zeitpunkten durch Einfügen oder Löschen von Zeilen erweitert oder reduziert werden (Letzteres nur, wenn die zu löschende Position im lfd. Jahr noch nicht belegt ist). Anders als in div. anderen Softwarelösungen ist die **Tabellenmodifikation auf Monatsbasis** möglich. Außerdem sind nachträgliche Änderungen der erfassten Daten zeitlich unbegrenzt möglich, einschließlich der beliebigen Häufigkeit von Teil-Erfassungen.

*Zurück zur Startdatei* kopiert den Dateinamen für die Zwischendatei aus dem Blatt *Stamm* C41 in das Blatt *Depot* A1 und speichert das Blatt *Depot* (ggf. mit Abfrage, ob belegt) als Datei:

```
Sheets("Depot").Copy  
Application.DisplayAlerts = False  
ActiveWorkbook.SaveAs Filename:=Cells(1, 1)  
Application.DisplayAlerts = True  
ActiveWorkbook.Close (False)
```

Anmerkung: Die **Erfassung** der Daten erfolgt durch Eingabe, **entweder** aus **Papier**-Tagesberichten **oder direkt** in der Produktion. Ggf. kann auch der Produktionsplan erfasst werden, mit anschließender Modifikation, sofern der Plan nicht exakt realisiert ist.

## Schritt 3 von 4: Monatsstatistik

Blattnamen: *Prod*, *EinProd*, *Quelle*, *Formel*, *Format*, *Stamm*

### Blatt Prod

	C	D	E	F	G	AK	AL
2	<b>Firmenname</b>		<b>Tab-Wechsel</b>		<b>Zurück zur Startdatei</b>		
3	<b>Produktion</b>				Menge in kg		
	<b>Zur Auswahl</b>				6.400		6.400
10	Nr.	Produkt	Einheit	kg/E	1.	31.	Gesamt
11	110	Produkt 110, 250g	Steige	5,00	6.400		6.400

Beim Aufruf der Datei aus der Startdatei wird der Stammbereich umkopiert und das Top-Makro *TopMonat* ausgelöst. Das Makro holt den Datenstand aus der Zwischendatei in das Blatt *Quelle*, kopiert die Kilo-Mengen in das Blatt *Prod*, formatiert die Tabelle ab Zeile 11 und setzt die Formeln für die Zeilen- und Spaltensummen ein (Formelhinterlegung im Blatt *Formel*).

Über *Tab-Wechsel* wird zwischen der Anzeige von Einheiten und Kilo gewechselt. Summe oben ist obligatorisch (entfällt bei Einheiten).

*Zur Auswahl* zeigt im Blatt *EinProd* die Daten des Produkts zur gewählten Cursor-Position:

	C	D	E
2	<b>Firmenname</b>		<b>Zurück</b>
3	<b>Produktion</b>		
5	<b>110</b>		
6	<b>Produkt 110, 250g</b>		
7	<b>Steige</b>		
9		<b>Menge</b>	
10	<b>Tag</b>	<b>Einheiten</b>	<b>kg</b>
11	1.	1.280	6.400
41	31.		
42	Ges.	1.280	6.400

Wichtige Makropassagen:

#### Zum Tab-Wechsel:

```
Sheets("Prod").Select
```

```
If Cells(3, 10) = "Menge in Einheiten" Then
```

```
Range("J3").Select
```

```
ActiveCell.FormulaR1C1 = "Menge in Kilo"
```

```
Sheets("Quelle").Select
```

```
Range("C11:D1010,F11:G1010,BS11:CW1010").Select
```

#### Else

```
Range("J3").Select
```

```
ActiveCell.FormulaR1C1 = "Menge in Einheiten"
```

```
Sheets("Quelle").Select
```

```
Range("C11:D1010,F11:G1010,S11:AW1010").Select
```

#### End If

```
Selection.Copy
```

‘ Daten in das Blatt *Prod* C11 einsetzen (Werte)

### Zur Auswahl:

Cursor auf eine Nr. in Spalte C setzen, Schaltfläche *Zur Auswahl* anklicken.

Cursorposition in das Blatt *Quelle* B1 kopieren.

Im Blatt *Formel* in B11 folgende Formel hinterlegen: =WENN(C11=C\$1;1;0)

Formel in das Blatt *Quelle* B11-B1010 kopieren, Werte fixieren.

B11-CW1010 absteigend B11 sortieren. Der zur gewählten Nr. passende Datensatz steht nun in Zeile 11 und kann in das Blatt *EinProd* kopiert werden.

### Zurück aus Blatt EinProd:

Entfernt die Daten in D11-E41 (*Selection.ClearContents*), wählt das Blatt *Prod*.

### Zurück zur Startdatei:

Löscht im Blatt *Prod* die Daten in C11-AL1010, wählt das Blatt *Prod*.

Löscht alle Daten im Blatt *Quelle*. Speichert die Datei. Öffnet die Startdatei.

## Schritt 4 von 4: Jahresstatistik

Blattnamen: *Prod*, *Quelle*, *Formel*, *Format*, *Stamm*

### Blatt Prod

	C	D	E	F	G	AK	AL
2	<b>Firmenname</b>					<b>Zurück zur Startdatei</b>	
3	<b>Produktion</b>				Menge in kg		
	<b>Tab-Wechsel</b>				6.400		6.400
10	Nr.	Produkt	Einheit	kg/E	Jan	Dez	Gesamt
11	110	Produkt 110, 250g	Steige	5,00	6.400		6.400

### Blatt Stamm:

	C	H
11	Firmenname	11 C:\Prod\zProd-1-2019.xlsx
12	August	12 C:\Prod\zProd-2-2019.xlsx
13	8	... usw. bis ...
14	2019	22 C:\Prod\zProd-12-2019.xlsx
21	C:\Prod	
31	C:\Prod\Start.xlsm	
32	C:\Prod\Erfassung.xlsm	
33	C:\Prod\Monat.xlsm	
34	C:\Prod\Jahr.xlsm	
41	C:\Prod\zProd-8-2019.xlsx	

Der Hinterlegungsbereich H11-H22 ist wie folgt auf die ab C11 einkopierten Daten verknüpft.

Beispiel: H11: =C21&"\zProd-1-"&C14&".xlsx"

## Wichtige Makropassagen:

### **Beim Öffnen der Datei aus der Startdatei:**

```
Sub TopJahr()  
  uQRJan  
  uQRFeb  
  ' ... usw. bis ...  
  uQRDez  
    uGemeinsameListe  
    uAuswertung  
End Sub  
  
Sub uQRJan()  
  Sheets("Stamm").Select  
  strPfad = Cells(11, 8)  
  If Dir$(strPfad, vbNormal) <> "" Then  
    Workbooks.Open Filename:=Cells(11, 8)  
  u1QR  
    Range("C1011").Select  
  u2QR  
  End If  
End Sub  
  
Sub uQRFeb()  
  Sheets("Stamm").Select  
  strPfad = Cells(12, 8)  
  If Dir$(strPfad, vbNormal) <> "" Then  
    Workbooks.Open Filename:=Cells(12, 8)  
  u1QR  
    Range("C2011").Select  
  u2QR  
  End If  
End Sub  
  
' ... usw. bis ...  
  
Sub uQRDez()  
  Sheets("Stamm").Select  
  strPfad = Cells(22, 8)  
  If Dir$(strPfad, vbNormal) <> "" Then  
    Workbooks.Open Filename:=Cells(22, 8)  
  u1QR  
    Range("C12011").Select  
  u2QR  
  End If  
End Sub  
  
Sub u1QR()  
  Range("C11:G1010,AX11:AX1010,CX11:CX1010").Select  
  Selection.Copy  
  Application.Windows(2).Activate  
  Sheets("Quelle").Select  
End Sub
```

```

Sub u2QR()
    Selection.PasteSpecial Paste:=xlPasteValues
    Application.CutCopyMode = False
    Range("A1").Select
    Application.Windows(2).Close (False)
End Sub

Sub uGemeinsameListe()
    Im Blatt Quelle: C1011:D13010,F1011:G13010 nach C21011 kopieren (Werte)-
    Aufsteigend C21011 sortieren. C21011:C33010 nach A21011 kopieren (Einfügeblock).
    Aus dem Blatt Formel A11 =WENN(C11<>C10;1;"a") in den Einfügeblock im Blatt Quelle
    kopieren, Werte fixieren. A21011:F33010 aufsteigend A21011 sortieren.
    In A21010 den Buchstaben a schreiben: ActiveCell.FormulaR1C1 = "a"
    Im belegten Block ab A21010 die Buchstaben a löschen:
    Selection.SpecialCells(xlCellTypeConstants, 2).Select
    If Cells(21011, 1) = 1 Then
        Range("A40000").Select
        Selection.End(xlUp).Select
        ActiveCell.Offset(1, 0).Select
        ActiveCell.EntireRow.Select
        Selection.Clear
        ActiveCell.Offset(1, 2).Select
        Selection.CurrentRegion.Select
        Selection.Clear
        Range("A21011").Select
        Selection.CurrentRegion.Select
        Selection.DataSeries Rowcol:=xlColumns, Type:=xlLinear, Date:=xlDay, _
            Step:=1, Trend:=False
        Application.CutCopyMode = False
    A21011:A22010,C21011:F22010 nach B11 kopieren (Werte).
    End If
    A21011:F33010 löschen
End Sub

```



Sub uAuswertung()

Aus dem Blatt *Formel* K11 =WENN(UND(\$B11<>0;\$C11=\$C10;H10<>0);H10;0) in das Blatt *Quelle* K5 kopieren (Formeln).

Spalte = 7

Zeile = 1011

While Zeile < 13011

If Cells(Zeile, 3) <> 0 Then

Range(Cells(Zeile, 3), Cells(Zeile + 999, 9)).Select

‘ nach C2011 kopieren (Werte)

B11:C1010 nach B21011 kopieren (Werte).

B20011:I22010 aufsteigend C20011 sortieren.

C2011;C22010 nach K20011-L20011 kopieren (Werte).

Formel aus K5 in den Block ab K20011 kopieren, Werte fixieren.

B20011:L22010 aufsteigend B20011 sortieren.

K20011:K21010 nach *Cells(11, Spalte).Select* kopieren (Werte).

L20011:L21010 nach *Cells(11, Spalte + 22).Select* kopieren (Werte).

B20011:L22010 löschen.

End If

Zeile = Zeile + 1000

Spalte = Spalte + 1

Wend

Aus dem Blatt *Quelle* C11:R1010 in das Blatt *Prod* C11 kopieren (Werte).

Im Blatt *Quelle* Summen rechts und Zeilenformat einsetzen.

In G3: *ActiveCell.FormulaR1C1 = "Menge in Einheiten"*

End Sub

### **Zum Tab-Wechsel:**

(analog wie bei der Monatsstatistik)

### **Zurück zur Startdatei:**

(analog wie bei der Monatsstatistik)

Anmerkungen: Die Jahresdatei wird bei jedem Aufruf komplett neu gefüllt und durchgerechnet. Dies dauert bei voller Auslastung mit 12 x 1.000 Datensätzen, welche jedes Mal auf eine gemeinsame Liste ausgerichtet werden müssen, nur **3 Sekunden**, weil ausschließlich mit einfachen Wenn-Abfragen gearbeitet wird.

Praxiserfahrungen: In den meisten in der Praxis anzutreffenden Excel-Lösungen fällt auf, dass diese auch bei Weiderholungsarbeiten im Direktverfahren strukturiert sind: man nehme eine Tabelle, schreibe etwas hinein, ein paar Formeln und Formate, fertig. In komplexeren Aufgaben entstehen Hunderte von Tabellen voller Formeln und Verknüpfungen, und viel unnötige Arbeitszeit geht mit Suchen im Explorer und der Tabellenpflege verloren. Unrationelle Arbeitsweisen sind also daran zu erkennen, dass mit Excel direkt gearbeitet wird. Sämtliche Wiederholungsarbeiten – und in der Statistik und Kalkulation ist dies der Hauptteil – sind voll automatisierbar, d. h. per Mausklick durchzuführen. In dem gezeigten Modell ist auch die zeitliche Fortschreibung, die sonst relativ viel Zeit in Anspruch nimmt, automatisiert. Man stellt per Schaltflächenklick den Monat ein, alles andere erledigt Excel selbst.

Unrationelle Zustände fallen häufig nicht auf, weil es gegenüber der vorhergehenden Arbeitsweise bereits einen Fortschritt darstellt. Im Prinzip hat man jedoch lediglich die Schreib- und Rechenmaschine in den PC verlagert. Wenigstens entfällt das lästige Kreuz- und Queraddieren. Viel mehr ist es aber nicht. Ein weiteres Hemmnis könnte sein, dass die Leute froh sind, wenn sie Arbeit haben. Unrationelles Arbeiten zur Sicherung von Arbeitsplätzen ist in unserem Wirtschaftssystem bis auf Weiteres jedoch keine gute Empfehlung. Ein hohes Rationalisierungspotential ist auch immer anzunehmen, wenn an Büro-Arbeitsplätzen in unmittelbarer Nähe zum Bildschirm ein Handrechner steht.

Die beschriebene Lösung ist in einer Molkerei seit 3 Jahren im täglichen Einsatz. Der Zeitaufwand konnte von zuvor 2 Stunden pro Tag auf eine halbe Stunde reduziert werden. Besonders positiv wurde aufgenommen, dass keine Dateien und Tabellen mehr angelegt und gepflegt werden müssen und die Listenanpassung durch Zeilen-Löschen oder -Einfügen **monatlich** möglich ist – ganz im Gegensatz zur sonstigen Üblichkeit, wegen der Jahresstatistik einmal angelegte Posten jahrelang durchschleppen zu müssen.

Das Produktionsprogramm umfasst ca. 150 Artikel, so dass eine Erfassung in der vorgestellten Form per Tastatur-Eingabe noch gut zu handhaben ist. In größeren Strukturen mit mehreren hundert Nummern wäre eine andere Lösung angesagt, in der Regel liegen dann die Daten bereits in Tabellenform vor. Eine Doppelerfassung kommt natürlich nicht in Frage, sondern wäre durch einen Kopiermodus zu ersetzen, welcher ebenfalls per Mausklick automatisiert werden kann. Die beschriebene Erfassung der Elementardaten über Papier-Tagesberichte gilt in der Lebensmittelbranche als nicht mehr zeitgemäße Methode, wird aber von einigen Unternehmen nach wie vor präferiert. Unabhängig davon stellt das vorgestellte Modell eine Vorlage für ähnlich gelagerte Aufgaben dar (z. B. Frachtkosten-Kontrolle, Energiestatistik, etc.).