

# Grundlagen der Entwicklung von Arbeitsblattanwendungen



## *In diesem Kapitel*

- ▶ Die grundlegenden Schritte bei der Entwicklung von Arbeitsblattanwendungen entdecken
  - ▶ Die Anforderungen des Endanwenders ermitteln
  - ▶ Anwendungen so planen, dass sie den Anforderungen der Anwender entsprechen
  - ▶ Anwendungen entwickeln und testen
  - ▶ Ihre Anwendungsentwicklung dokumentieren und die Dokumentation für die Anwender schreiben
- 

## *Was ist eine Arbeitsblattanwendung*

Im Rahmen dieses Buches ist eine Arbeitsblattanwendung eine Excel-Arbeitsmappe (oder eine Gruppe von zusammengehörenden Dateien), die so entwickelt wurde, dass auch jemand anders als der Programmierer ohne viel Lernaufwand nützliche Arbeiten erledigen kann. Legt man diese Definition zugrunde, dann sind vermutlich die meisten Arbeitsmappen, die Sie entwickelt haben, keine Arbeitsblattanwendungen. Vermutlich befinden sich auf Ihrer Festplatte Dutzende oder Hunderte Excel-Arbeitsmappen, und vermutlich sind wir auf der sicheren Seite, wenn wir wetten, dass die meisten davon nicht so entworfen sind, dass sie von anderen genutzt werden können.

Eine gute Arbeitsblattanwendung:

- ✓ Ermöglicht es dem Endanwender, eine Aufgabe durchzuführen, die er anderenfalls nicht erledigen könnte.
- ✓ Stellt für ein Problem eine angemessene Lösung bereit. (Eine Arbeitsblattumgebung ist nicht immer der optimale Ansatz.)
- ✓ Erledigt das, was sie erledigen soll. Diese Anforderung mag selbstverständlich erscheinen, jedoch ist es nicht untypisch, dass Anwendungen diesen Test nicht bestehen.
- ✓ Erzeugt saubere Ergebnisse und enthält keine Fehler.
- ✓ Verwendet geeignete und effiziente Methoden und Algorithmen, um die Aufgabe zu erledigen.
- ✓ Fängt Fehler früh genug ab, damit sich der Endanwender nicht damit befassen muss.
- ✓ Erlaubt es dem Anwender nicht, wichtige Komponenten zu verändern oder zu löschen (und zwar weder unbeabsichtigt noch beabsichtigt).

- ✓ Besitzt eine deutliche und konsistente Benutzeroberfläche, damit der Anwender immer genau weiß, was er als Nächstes machen muss.
- ✓ Zeichnet sich durch eine gute Dokumentation der Formeln, Makros und Benutzeroberflächenelemente aus, damit im Bedarfsfall auch später noch Änderungen vorgenommen werden können.
- ✓ Ist so entworfen, dass sie ohne allzu große Änderungen einfach modifiziert werden kann. Es ist eine Grundgegebenheit, dass sich die Anforderungen der Anwender im Laufe der Zeit ändern.
- ✓ Besitzt ein Hilfesystem, auf das einfach zugegriffen werden kann und in dem sich zumindest für die wichtigsten Prozeduren nützliche Informationen befinden.
- ✓ Ist so entworfen, dass sie portabel ist und auf jedem System läuft, auf dem die betreffende Software (in unserem Fall eine geeignete Kopie von Excel) vorhanden ist.

Es ist vermutlich nicht verwunderlich, dass es möglich ist, Arbeitsblattanwendungen für unterschiedliche Aufgaben und unterschiedlicher Komplexität zu erstellen, angefangen von einer einfachen Vorlage, bei der lediglich bestimmte Zellen ausgefüllt werden müssen, bis zu einer extrem umfangreichen Anwendung, die eine spezielle Benutzeroberfläche verwendet, die vielleicht gar nicht mehr wie ein Tabellenblatt aussieht.

## ***Schritte bei der Entwicklung der Anwendung***

Es gibt kein einfaches, narrensicheres Rezept, um eine effiziente Arbeitsblattanwendung zu entwickeln. Jeder hat seinen eigenen Stil und eine eigene Vorgehensweise, um solch eine Anwendung zu entwickeln. Außerdem ist jedes Projekt anders und erfordert daher einen individuellen Ansatz. Und schließlich spielen auch die Anforderungen und der technische Kenntnisstand derjenigen, mit denen oder für die Sie arbeiten, eine Rolle dabei, wie der Entwicklungsprozess verläuft.

Entwickler von Arbeitsblattanwendungen führen in der Regel die folgenden Aktivitäten durch:

- ✓ Die Anforderungen der Anwender ermitteln
- ✓ Eine Anwendung planen, die diese Anforderungen erfüllt
- ✓ Die am besten geeignete Benutzeroberfläche festlegen
- ✓ Das Arbeitsblatt, die Formeln, die Makros und die Benutzeroberfläche erstellen
- ✓ Die Anwendung testen und debuggen
- ✓ Versuchen, die Anwendung unverwundbar zu machen
- ✓ Die Anwendung ästhetisch ansprechend und intuitiv gestalten
- ✓ Den Entwicklungsprozess dokumentieren
- ✓ Die Endbenutzerdokumentation und das Hilfesystem erstellen
- ✓ Die Anwendung an die Benutzer verteilen
- ✓ Die Anwendung aktualisieren, sollte dies erforderlich sein

Beachten Sie, dass all diese Schritte bei allen Anwendungen erforderlich sind. Die Reihenfolge, in denen diese Aktivitäten durchgeführt werden, kann von Projekt zu Projekt variieren. Wir beschreiben die einzelnen Aktivitäten auf den weiteren Seiten dieses Kapitels. Für die meisten Elemente werden die technischen Details in den nachfolgenden Kapiteln beleuchtet.

## ***Die Anforderungen der Anwender ermitteln***

Wenn Sie sich an ein neues Excel-Projekt machen, besteht einer der ersten Schritte darin, genau zu verstehen, was die Endanwender wollen und brauchen. Falls es versäumt wird, die Anforderungen der Anwender bereits zu einem frühen Zeitpunkt festzustellen, führt dies später oft zu zusätzlicher Arbeit, weil Sie die Anwendung so anpassen müssen, damit sie letztendlich das erfüllt, was sie bereits von Anfang an hätte erfüllen sollen.

In bestimmten Fällen kann es auch sein, dass Sie Ihre Endanwender genau kennen – vielleicht, weil Sie selbst zu dieser Gruppe gehören. In anderen Fällen (weil Sie beispielsweise ein externer Berater sind, der eine Anwendung für einen neuen Kunden entwickelt) wissen Sie vermutlich nur wenig über die Endanwender oder deren Arbeitsumgebung.

Wie stellen Sie die Anforderungen der Endanwender fest? Falls Sie mit der Entwicklung einer Arbeitsblattanwendung beauftragt sind, ist es eine gute Idee, sich mit den Endanwendern zu treffen und genaue Fragen zu stellen. Viel besser ist es, alles aufzuschreiben, Flussdiagramme zu erstellen, auch auf kleine Details zu achten und überhaupt alles zu tun, was sicherstellt, dass das Produkt, das Sie liefern, das Produkt ist, das gebraucht wird.

Hier sind ein paar Richtlinien, die helfen, die Phase zu vereinfachen:

- ✓ Gehen Sie nicht davon aus, dass Sie wissen, was der Anwender braucht. Falsche Vermutungen in dieser Phase führen später fast immer zu Problemen.
- ✓ Sprechen Sie, falls möglich, direkt mit den Benutzern der Anwendung, und nicht mit einem Supervisor oder Manager.
- ✓ Versuchen Sie, in Erfahrung zu bringen, was derzeit getan wird, um die Anforderungen der Anwender zu erfüllen (falls es so etwas gibt). Vielleicht können Sie etwas Arbeit einsparen, indem Sie eine vorhandene Anwendung modifizieren. Auf jeden Fall hilft es, sich vorhandene Lösungen anzusehen, weil Sie sich so mit den aktuellen Arbeitsabläufen vertraut machen können.
- ✓ Ermitteln Sie, welche Ressourcen in der Arbeitsumgebung der Anwender vorhanden sind. Bringen Sie beispielsweise in Erfahrung, ob Sie bestimmte Hardware- oder Softwarebeschränkungen berücksichtigen müssen.
- ✓ Versuchen Sie, falls möglich, herauszufinden, welche Hardware-Systeme verwendet werden. Falls Ihre Anwendung auf langsamen Systemen verwendet werden wird, müssen Sie dies berücksichtigen.
- ✓ Untersuchen Sie, welche Excel-Versionen eingesetzt werden. Obwohl Microsoft alles tut, was im Rahmen seiner Möglichkeit liegt, die Anwender zum Update auf die aktuelle Version zu bewegen, tut dies der Mehrheit der Excel-Anwender nicht.
- ✓ Stellen Sie fest, welche Fähigkeiten und Kenntnisse die Endanwender besitzen. Diese Informationen unterstützen Sie dabei, die passende Anwendung zu entwickeln.

- ✓ Stellen Sie fest, über welchen Zeitraum die Anwendung eingesetzt werden soll und ob während der Lebensdauer des Projekts irgendwelche Änderungen oder Anpassungen zu erwarten sind. Die Kenntnis dieser Informationen hilft dabei einzuschätzen, wie viel Aufwand Sie in die Anwendung stecken sollten, und außerdem können Sie dann bereits in einem frühen Stadium Änderungen einplanen.

Zum guten Schluss: Seien Sie nicht überrascht, wenn sich die Projektspezifikationen ändern, bevor Sie die Anwendung fertiggestellt haben. Dies passiert fast immer, und Sie befinden sich in einer besseren Position, wenn Sie diese Änderungen erwarten, als wenn Sie von ihnen überrascht werden. Stellen Sie sicher, dass Ihr Vertrag (falls es einen gibt) das Problem einer sich ändernden Spezifikation berücksichtigt.

## ***Eine Anwendung planen, die die Anforderungen der Anwender erfüllt***

Nachdem Sie die Anforderungen der Anwender ermittelt haben, ist die Versuchung oft groß, einfach loszulegen und in Excel herumzuprobieren. Nehmen Sie einen Rat von jemandem an, der genau unter diesem Problem gelitten hat: Halten Sie sich ein wenig zurück. Baufacharbeiter bauen kein Haus, ohne dass sie hierfür eine Zeichnung besitzen. Und Sie sollten keine Arbeitsblattanwendung entwickeln, ohne vorher einen Plan gemacht zu haben. Wie formal Ihr Plan aussieht, hängt vom Umfang Ihres Projekts und davon ab, wie Sie normalerweise arbeiten. Dennoch sollten Sie auf jeden Fall vor der Umsetzung ein wenig darüber nachdenken, was Sie machen wollen, und hierfür einen Aktionsplan erstellen.

Bevor Sie nun die Ärmel hochkrempeln und sich vor Ihre Tastatur begeben, sollten Sie sich ein wenig Zeit nehmen und die verschiedenen Ansätze berücksichtigen, mit denen Sie das Problem lösen können. Gründliche Excel-Kenntnisse zählen sich in dieser Planungsphase aus. Es ist immer besser, Sackgassen zu vermeiden, als in sie hineinzustolpern.

Wenn Sie ein Dutzend Excel-Experten bitten, anhand von präzisen Anforderungen eine Excel-Anwendung zu entwickeln, erhalten Sie ein Dutzend unterschiedlicher Implementierungen dieses Projekts, die alle diesen Anforderungen genügen. Manche dieser Lösungen werden besser sein als andere, da Excel unterschiedliche Möglichkeiten zur Verfügung stellt, eine bestimmte Aufgabe zu erledigen. Wenn Sie Excel in- und auswendig kennen, dann haben Sie eine gute Vorstellung über die potenziellen Ansätze, die Ihnen zur Verfügung stehen, und Sie können sich dann für denjenigen entscheiden, der Ihnen für das jeweilige Projekt am besten geeignet erscheint. Manchmal hilft auch ein wenig Querdenken, da Sie so zu einem eher unüblichen Ansatz kommen können, der dennoch besser ist als die, die im ersten Moment auf der Hand zu liegen scheinen.

Am Anfang dieser Planungsphase sollten Sie sich ein paar allgemeinere Fragen stellen. Hierzu gehören:

- ✓ **Dateistruktur:** Denken Sie darüber nach, ob Sie eine Arbeitsmappe mit mehreren Arbeitsblättern, mehrere Arbeitsmappen mit jeweils einem Arbeitsblatt oder ob Sie eine Vorlagendatei verwenden wollen.

- ✓ **Datenstruktur:** Sie sollten immer bedenken, wie Sie die Daten strukturieren wollen, und hierbei ebenfalls bedenken, ob Sie zur Datenspeicherung externe Datenbankdateien verwenden oder ob Sie alles in Arbeitsblättern speichern wollen.
- ✓ **Add-In oder Arbeitsmappendatei:** In manchen Fällen kann ein Add-In die beste Wahl für das fertige Produkt sein. Oder vielleicht verwenden Sie eine Kombination aus Add-In und einer Standardarbeitsmappe.
- ✓ **Excel-Version:** Wird Ihre Anwendung nur mit Excel 2016 verwendet? Mit Excel 2010 oder neuer? Was ist mit Excel 2003 und älteren Versionen? Soll Ihre Anwendung auch auf dem Mac laufen? Diese Überlegungen sind wichtig, da jede neue Excel-Version Features enthält, die in den Vorgängerversionen nicht verfügbar sind. Die Menübandoberfläche, die in Excel 2007 eingeführt wurde, macht es deutlich schwieriger, eine Anwendung zu erstellen, die auch mit älteren Excel-Versionen funktioniert.
- ✓ **Fehlerbehandlung:** Ein wichtiges Thema für jede Anwendung ist die Fehlerbehandlung. Sie müssen festlegen, wie Ihre Anwendung Fehler entdeckt und wie sie damit umgeht. Falls Ihre Anwendung beispielsweise das aktive Arbeitsblatt formatiert, müssen Sie auch mit dem Fall umgehen können, dass ein Diagrammblatt und kein Tabellenblatt aktiv ist.
- ✓ **Verwendung spezieller Features:** Falls Ihre Anwendung umfangreiche Daten zusammenfassen soll, erwägen Sie vielleicht, das Excel-Feature PivotTables zu verwenden. Oder Sie setzen das Feature Datenüberprüfung ein, um die Gültigkeit der eingegebenen Daten zu überprüfen.
- ✓ **Geschwindigkeit:** Der richtige Zeitpunkt, um über die Optimierung der Geschwindigkeit und die Effizienz Ihrer Anwendung nachzudenken, ist die Entwicklungsphase und nicht erst dann, wenn die Anwendung fertiggestellt ist und Anwender sich beschweren, dass sie träge reagiert.
- ✓ **Sicherheitsstufen:** Wie Sie vielleicht wissen, stellt Excel verschiedene Schutzoptionen bereit, mit denen der Zugriff auf bestimmte Elemente einer Arbeitsmappe eingeschränkt werden kann. So können Sie beispielsweise Zellen sperren, damit die enthaltenen Formeln nicht geändert werden können. Sie können auch Kennwörter vergeben und so verhindern, dass unberechtigte Personen bestimmte Dateien ansehen oder öffnen. Wenn Sie im Vorhinein genau festlegen, was Sie schützen müssen und welche Sicherheitsstufen nötig sind, machen Sie sich Ihr Leben leichter.



Bitte denken Sie daran, dass die Schutzmechanismen von Excel nicht 100 %ig sicher sind; weit davon entfernt. Falls Sie für Ihre Anwendung vollständige und absolute Sicherheit benötigen, dann ist Excel möglicherweise nicht die geeignete Plattform.

In dieser Phase müssen Sie vermutlich viele weitere projektspezifische Überlegungen anstellen. Ziehen Sie alle Optionen in Erwägung und verwenden Sie nicht sofort die erste Lösung, die Ihnen in den Sinn kommt.

Bei der Designphase sollten Sie immer auch zukünftige Änderungen an Ihrer Anwendung im Auge behalten. Tun Sie sich selbst einen Gefallen und entwerfen Sie Ihre Anwendung so allgemein wie möglich. Schreiben Sie also beispielsweise keine Prozedur, die nur mit einem ganz

bestimmten Zellbereich funktioniert. Erstellen Sie stattdessen eine Prozedur, der ein Zellbereich als Argument übergeben werden kann. Falls die unvermeidlichen Änderungen gefordert werden, erleichtert ein solches Design, die Bearbeitungen vorzunehmen. Außerdem werden Sie oft feststellen, dass die Arbeiten, die Sie für ein Projekt vornehmen, denen ähnlich sind, die Sie bei einem anderen Projekt erledigen. Behalten Sie die Wiederverwendbarkeit Ihres Codes im Hinterkopf, wenn Sie ein Projekt planen.

Vermeiden Sie es, sich bei der Lösung eines Problems komplett vom Endanwender leiten zu lassen. Nehmen Sie beispielsweise an, dass Sie sich mit einem Manager treffen, der Ihnen erzählt, dass eine Abteilung eine Anwendung benötigt, die Textdateien erstellt, die dann in eine andere Anwendung importiert werden sollen. Setzen Sie die Anforderungen der Anwender nicht mit der Lösung der Aufgabe gleich. Die tatsächliche Anforderung der Anwender besteht darin, Daten auszutauschen. Die Verwendung einer Textdatei als Zwischenglied ist nur ein möglicher Ansatz zur Lösung dieser Aufgabe; möglicherweise existieren andere Lösungen, die besser geeignet sind. Anders ausgedrückt: Lassen Sie die Anwender ihr Problem nicht dadurch definieren, dass sie einen Lösungsansatz vorschlagen. Es ist *Ihre* Aufgabe, den besten Lösungsansatz zu entwickeln.

## ***Die am besten geeignete Benutzeroberfläche festlegen***

Wenn Sie Arbeitsblätter entwickeln, die von anderen eingesetzt werden, müssen Sie insbesondere auf die Gestaltung der Benutzeroberfläche achten. Unter *Benutzeroberfläche* verstehen wir die Verfahren, mit denen der Anwender mit Ihrer Anwendung interagiert und Ihre VBA-Makros ausführt.

Seit der Einführung von Excel 2007 sind einige der Überlegungen zur Gestaltung der Benutzeroberfläche nicht mehr relevant. Benutzerdefinierte Menüs und Symbolleisten lassen sich ab Excel 2007 nicht mehr verwenden. Als Folge daraus müssen Entwickler lernen, mit dem Menüband umzugehen.

Excel stellt verschiedene Features zur Verfügung, die für die Gestaltung der Benutzeroberfläche wichtig sind:

- ✓ Anpassen des Menübands
- ✓ Anpassen der Kontextmenüs
- ✓ Tastenkombinationen
- ✓ Benutzerdefinierte Dialogfelder (UserForms)
- ✓ Steuerelemente (wie ein Listefeld oder eine Schaltfläche), die direkt auf dem Arbeitsblatt abgelegt sind

Wir stellen diese Features in den folgenden Abschnitten kurz vor und behandeln sie in späteren Kapiteln ausführlicher.

## Anpassen des Menübands

Das Menüband, das in Excel 2007 eingeführt wurde, stellt einen gravierenden Paradigmenwechsel bei der Gestaltung der Benutzeroberfläche dar. Glücklicherweise haben Entwickler ziemlich viel Kontrolle über das Menüband. Obwohl Excel es den Endanwendern erlaubt, das Menüband anzupassen, so stellen Änderungen an der Benutzeroberfläche vom Code aus keine einfache Aufgabe dar.



In Kapitel 17 finden Sie ausführliche Informationen über das Arbeiten mit dem Menüband.

## Anpassen der Kontextmenüs

Excel erlaubt es VBA-Entwicklern, die Kontextmenüs anzupassen, die sich nach einem Rechtsklick öffnen. Kontextmenüs stellen Anwendern eine komfortable Möglichkeit zur Verfügung, die gewünschte Aktion auszulösen, ohne sich hierbei zu sehr von dem Zellbereich wegzubewegen, in dem sie gerade arbeiten. In Abbildung 1.1 sehen Sie ein benutzerdefiniertes Kontextmenü, das angezeigt wird, nachdem eine Zelle mit der rechten Maustaste angeklickt wurde.



Abbildung 1.1: Ein angepasstes Kontextmenü



Kapitel 18 beschreibt, wie Sie von VBA-Code aus mit Kontextmenüs arbeiten. Sie lernen dort auch einige der Einschränkungen kennen, die sich aus der in Excel 2013 eingeführten Einzeldokumentschnittstelle (*Single Document Interface*, SDI) ergeben.

## Tastenkombinationen definieren

Eine weitere Gestaltungsmöglichkeit für die Benutzerschnittstelle ist eine benutzerdefinierte Tastenkombination. Sie können in Excel einem Makro eine Tastenkombination mit der **[Strg]**-Taste oder den Tasten **[Strg] + [F4]** zuweisen. Wenn der Anwender diese Tastenkombination drückt, wird das betreffende Makro ausgeführt.

Achten Sie jedoch auf mögliche Stolperfallen. Machen Sie dem Anwender deutlich, welche Tastenkombinationen aktiv sind und was sie machen. Eine Tastenkombination, die Sie einem Makro zuweisen, hat Vorrang vor den eingebauten Tastenkombinationen. So ist beispielsweise **Strg+S** eine Excel-Tastenkombination für das Speichern der aktuellen Datei. Falls Sie diese Tastenkombination einem Makro zuweisen, verlieren Sie die Möglichkeit, mit **Strg+S** die Datei zu speichern. Bedenken Sie, dass Tastenkombinationen die Groß- und Kleinbuchstaben unterscheiden, Sie können also auch Tastenkombinationen wie **Strg+⇧+S** verwenden.

## Benutzerdefinierte Dialogfelder erstellen

Jeder, der auch nur kurze Zeit mit einem PC gearbeitet hat, kennt mit Sicherheit die Dialogfelder. Daher können auch in Excel die Dialogfelder, die Sie für Ihre Anwendung entwerfen, eine wichtige Rolle spielen. Abbildung 1.2 zeigt ein Beispiel eines benutzerdefinierten Dialogfelds.

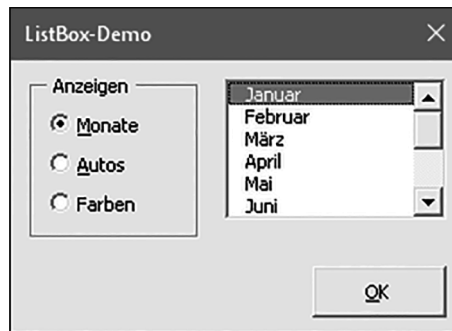


Abbildung 1.2: Ein Dialogfeld, das mit dem Excel-Feature UserForm erstellt wurde

Ein benutzerdefiniertes Dialogfeld wird in Excel *UserForm* genannt. Ein UserForm kann Eingaben anfordern, der Anwender kann hier Optionen und andere Einstellungen vornehmen und hiermit den Ablauf der gesamten Anwendung steuern. Die Elemente, die sich auf einem UserForm befinden (Schaltflächen, Dropdown-Listenfelder, Kontrollkästchen und so weiter), werden Steuerelemente genannt, genauer *ActiveX-Steuerelemente*. Excel stellt eine Vielzahl von ActiveX-Steuerelementen bereit; außerdem können Sie auch Steuerelemente von Drittanbietern integrieren.

Nachdem Sie ein Steuerelement in ein Dialogfeld eingefügt haben, können Sie es mit einer Zelle auf einem Arbeitsblatt verknüpfen, sodass es keine Makros benötigt (bis auf ein einfaches Makro, das das Dialogfeld anzeigt). Ein Steuerelement mit einer Zelle zu verknüpfen, ist zwar sehr einfach, jedoch ist dies nicht immer der beste Weg, um Benutzereingaben mit einem Dialogfeld abzufragen. In den meisten Fällen werden Sie VBA-Makros entwickeln, die mit Ihren benutzerdefinierten Dialogfeldern zusammenarbeiten.



Die Details zu UserForms werden in Teil III vorgestellt.



## ActiveX-Steuerelemente auf einem Arbeitsblatt verwenden

Sie können in Excel auch ActiveX-Steuerelemente auf die Zeichnungsebene eines Arbeitsblatts einfügen. Die Zeichnungsebene ist eine unsichtbare Ebene auf dem Arbeitsblatt, auf dem sich Bilder, Diagramme und andere Objekte befinden. Abbildung 1.3 zeigt ein einfaches Modell eines Arbeitsblatts, bei dem verschiedene UserForm-Steuerelemente direkt auf das Arbeitsblatt eingefügt wurden. Dieses Blatt enthält die folgenden ActiveX-Steuerelemente: ein Kontrollkästchen, eine Bildlaufleiste sowie zwei Gruppen mit Optionsfeldern. Diese Arbeitsmappe verwendet keine Makros; die Steuerelemente sind direkt mit Zellen des Arbeitsblatts verknüpft.



Diese Arbeitsmappe finden Sie in den Beispieldateien zu Kapitel 13 dieses Buches. Ihr Name lautet `Arbeitsblatt Steuerelemente.xlsx`.

	A	B	C	D	E	F	G	H	I	J
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										

### Hypothekenparameter

**Kaufpreis:**

☒ Finanzierung der Hypothek von \$5.000

**Rückzahlung:**

☐ 10%

☐ 15%

☒ 20%

**Laufzeit:**

☒ 30 Jahre fest

☐ 15 Jahre fest

**Hypothekenbetrag:**

**Monatliche Zahlung:**

[Amortisierungszeitplan](#)

### Verknüpfte Zelle

565 Zinsen  
5,65 Prozent

WAHR	Gebühren
\$281.000	Hypothek

WAHR	30 Jahre
FALSCH	15 Jahre
30 Jahre Laufzeit	

FALSCH	10% zurück
FALSCH	15% zurück
WAHR	20% zurück
20% Rückzahlung	

Wählen Sie Entwicklertools - Steuerelemente - Entwurfsmodus.  
Anschließend Entwicklertools - Steuerelemente - Eigenschaften  
für ein Steuerelement. Für die Verwendung den Entwurfsmodus beenden.

Abbildung 1.3: Sie können UserForm-Steuerelemente in ein Arbeitsblatt einfügen und diese mit Zellen verknüpfen.

Das vermutlich am häufigsten verwendete Steuerelement ist die Befehlsschaltfläche (*CommandButton*). Für sich alleine macht eine Befehlsschaltfläche nichts; daher müssen Sie der Schaltfläche ein Makro zuweisen.

Durch die Verwendung von Steuerelementen auf einem Arbeitsblatt ist es häufig nicht mehr erforderlich, benutzerdefinierte Dialogfelder zu erstellen. Die Bedienung eines Arbeitsblatts lässt sich oft dadurch vereinfachen, dass Sie in das Blatt ein paar ActiveX-Steuerelemente (oder Formularsteuerelemente) einfügen. Dank dieser ActiveX-Steuerelemente kann der Anwender mit vertrauten Steuerelementen Eingaben vornehmen und braucht diese nicht in eine Zelle einzugeben.

Sie finden diese Steuerelemente über den Befehl ENTWICKLERTOOLS|STEUERELEMENTE|EINFÜGEN (siehe Abbildung 1.4). Falls die Registerkarte ENTWICKLERTOOLS nicht im Menüband sichtbar ist, aktivieren Sie sie auf der Seite MENÜBAND ANPASSEN des Dialogfelds EXCEL-OPTIONEN.

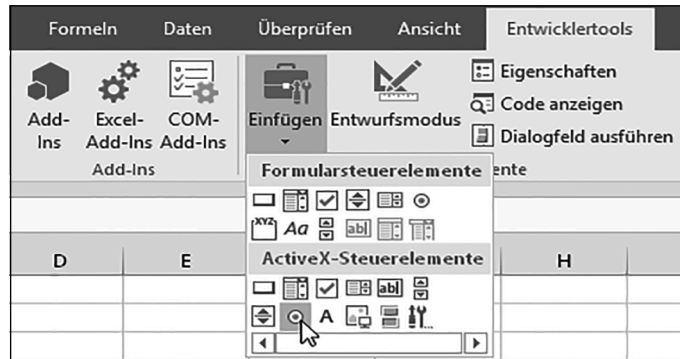


Abbildung 1.4: Verwenden Sie das Menüband, um Steuerelemente in ein Arbeitsblatt einzufügen.

Die Steuerelemente stehen in zwei verschiedenen Typen bereit: als Formularsteuerelemente und als ActiveX-Steuerelemente. Beide Steuerelementtypen haben ihre Vor- und Nachteile. Im Allgemeinen sind Formularsteuerelemente einfacher zu verwenden, jedoch sind ActiveX-Steuerelemente flexibler. Tabelle 1.1 stellt die beiden Arten der Steuerelemente gegenüber.

	ActiveX-Steuerelemente	Formularsteuerelemente
Excel-Versionen	97, 2000, 2002, 2003, 2007, 2010, 2013, 2016	5, 95, 97, 2000, 2002, 2003, 2007, 2010, 2013, 2016
Verfügbare Steuerelemente	Kontrollkästchen, Textfeld, Befehlsschaltfläche, Optionsfeld, Listefeld, Kombinationsfeld, Umschaltfläche, Drehfeld, Bildlaufleiste, Bezeichnungsfeld, Bild (und weitere, die hinzugefügt werden können)	Gruppenfeld, Schaltfläche, Kontrollkästchen, Optionsfeld, Listefeld, Kombinationsfeld, Bildlaufleiste, Drehfeld
Speicherort für Makros	Im Codemodul des Arbeitsblatts	In jedem Standard-VBA-Modul
Makroname	Korrespondiert mit dem Namen des Steuerelements (zum Beispiel <code>CommandButton1_Click</code> )	Beliebigen Namen, den Sie frei vergeben können
Entspricht	UserForm-Steuerelementen	Dialogsheet-Steuerelementen aus Excel vor Version 97
Anpassung	Umfassend, über das Eigenschaftsfenster	Minimal
Reagiert auf Ereignisse	Ja	Nur auf Click- und Change-Ereignisse

Tabelle 1.1: ActiveX-Steuerelemente vs. Formularsteuerelemente

## ***Die Anwendung entwickeln***

Nachdem Sie die Anforderungen der Anwender identifiziert haben, den Ansatz ausgewählt haben, den Sie verwenden wollen, um diese Anforderungen zu erfüllen, und entschieden haben, welche Komponenten Sie für die Benutzeroberfläche verwenden wollen, ist es an der Zeit, sich an das Wesentliche zu machen und die Anwendung zu entwickeln. Dieser Schritt umfasst selbstverständlich die meiste Zeit, die Sie für ein bestimmtes Projekt aufbringen.

Wie genau Sie bei der Entwicklung der Anwendung vorgehen, hängt sowohl von Ihrem individuellen Stil als auch von der Art der Anwendung ab. Sieht man von einfachen Arbeitsmappenvorlagen ab, bei denen lediglich die leeren Zellen gefüllt werden müssen, wird Ihre Anwendung vermutlich Makros einsetzen. In Excel Makros zu erstellen, ist einfach, *gute* Makros zu erstellen, ist jedoch schwierig.

## ***Denken Sie immer an den Anwender***

In diesem Abschnitt diskutieren wir wichtige Fragen bei der Entwicklung, die vor allem dann ans Tageslicht treten, wenn Ihre Anwendung mehr und mehr nutzbar wird und wenn sich der Zeitpunkt nähert, an dem Sie die fertige Anwendung ausliefern können.

## ***Die Anwendung testen***

Wie oft haben Sie schon eine kommerzielle Anwendung benutzt, die in einem entscheidenden Moment einfach abstürzt. Es ist sehr wahrscheinlich, dass die Ursache hierfür in ungenügenden Tests liegt, durch die nicht alle Softwarefehler gefunden und behoben wurden. Alle nicht triviale Software enthält Bugs, jedoch sind diese in der besseren Software eher verdeckt und schwer zu finden. Wie Sie noch sehen werden, müssen Sie auch in Excel ab und zu Workarounds für Bugs verwenden, damit Ihre Anwendung richtig funktioniert.

Nachdem Sie Ihre Anwendung erstellt haben, müssen Sie sie testen. Das Testen ist der entscheidende Schritt; es ist nicht unüblich, dass für das Testen und das Debuggen genau so viel Zeit aufgewendet wird wie für das Erstellen der Anwendung. Im Grunde sollten Sie bereits während der Entwicklungsphase einen großen Teil der Tests durchführen. Gleichgültig, ob Sie eine VBA-Routine programmieren oder ob Sie Formeln in einem Arbeitsblatt erstellen: Sie wollen auf jeden Fall, dass die Anwendung so funktioniert, wie es beabsichtigt ist.

So wie Standardanwendungen, die kompiliert werden, so sind auch Arbeitsblattanwendungen, die Sie erstellen, fehleranfällig. Ein Softwarebug lässt sich erstens als etwas definieren, das passiert, aber nicht passieren soll, während ein Programm oder eine Anwendung ausgeführt wird, und zweitens als etwas, das nicht passiert, obwohl es passieren sollte. Beide Bugtypen sind gleichermaßen unangenehm. Sie sollten daher einplanen, dass Sie einen Großteil der Entwicklungszeit damit verbringen, Ihre Anwendung unter allen möglichen Bedingungen zu testen und alle Probleme zu beheben, die Sie finden.

Es ist wichtig, jede Arbeitsblattanwendung, die Sie für andere entwickeln, gründlich zu testen. In Abhängigkeit von der genauen Zielgruppe müssen Sie die Anwendung auch narrensicher gestalten. Anders ausgedrückt: Versuchen Sie, alle möglichen Fehler und Pannen, die

eventuell auftreten können, zu antizipieren, und versuchen Sie dann, sie zu vermeiden oder zumindest elegant zu behandeln. Diese Weitsicht hilft nicht nur den Nutzern, sondern sie macht es auch Ihnen leichter und dient Ihrer Reputation. Erwägen Sie ebenfalls, einen Beta-test durchzuführen; Ihre Endanwender sind hierfür geeignete Kandidaten, da sie auch diejenigen sein werden, die Ihr Produkt später nutzen. (Siehe auch den Kasten mit dem Titel »Wie wäre es mit einem Betatest?« etwas weiter hinten.)

Obleich Sie nicht alle denkbaren Möglichkeiten testen können, so sollten Ihre Makros doch die häufig vorkommenden Fehler behandeln. Was soll beispielsweise passieren, wenn der Anwender statt eines numerischen Werts eine Zeichenfolge eingibt? Was, wenn der Anwender versucht, ein Makro zu starten, während keine Arbeitsmappe geöffnet ist. Was, wenn er ein Dialogfeld abbricht, ohne eine Auswahl zu treffen? Was passiert, wenn der Anwender Strg+F6 drückt und zum nächsten Fenster springt? Je mehr Erfahrungen Sie sammeln, desto vertrauter werden Sie mit dieser Art von Problem und Sie werden sie berücksichtigen, ohne viel darüber nachzudenken.

### ***Wie wäre es mit einem Betatest?***

Softwarehersteller haben in der Regel einen rigorosen Testzyklus für neue Produkte. Nach ausführlichen internen Tests wird eine Vorabversion der Software einer Gruppe interessierter Anwender für einen Betatest zur Verfügung gestellt. In dieser Phase werden oft weitere Probleme aufgedeckt, die normalerweise noch vor der endgültigen Freigabe und Veröffentlichung der Software behoben werden.

Wenn Sie eine Excel-Anwendung entwickeln, die von mehr als nur ein paar Leuten eingesetzt wird, dann sollten Sie erwägen, einen Betatest durchzuführen. Durch diesen Test können Teile der beabsichtigten Nutzergruppe Ihre Anwendung in der angestrebten Umgebung auf (meistens) unterschiedlicher Hardware einsetzen.

Die Betaphase sollte beginnen, nachdem Sie Ihre eigenen Tests abgeschlossen und das Gefühl haben, dass Ihre Anwendung verteilt werden kann. Sie müssen eine Gruppe von Anwendern identifizieren, von der Sie denken, dass sie Ihnen helfen kann. Der Prozess verläuft dann optimal, wenn Sie all das verteilen, was letztendlich Bestandteil Ihrer Anwendung sein soll: Anwenderdokumentation, Installationsprogramm, Hilfedateien und so weiter. Sie können den Betatest auf unterschiedliche Arten auswerten: Gespräche von Angesicht zu Angesicht oder per Telefon, E-Mails und Umfragen.

Fast immer stoßen Sie so auf Probleme, die Sie beheben müssen, oder Sie hören Verbesserungsvorschläge, bevor Sie Ihre Anwendung ausliefern. Natürlich benötigt die Betaphase zusätzliche Zeit, ein Luxus, den sich nicht alle Projekte leisten können.

### ***Die Anwendung narrensicher machen***

Wenn Sie darüber nachdenken, dann ist es ziemlich einfach, ein Arbeitsblatt unbrauchbar zu machen. Das Löschen einer wichtigen Formel oder eines Werts kann zu Fehlern im gesamten Arbeitsblatt und vielleicht auch in abhängigen Blättern führen. Es wird noch schlimmer, wenn die zerstörte Arbeitsmappe gespeichert wird und so die noch funktionierende Kopie auf

der Festplatte überschreibt. Falls dann noch kein Backup vorhanden ist, steckt der Nutzer Ihrer Anwendung in Problemen; möglicherweise werden Sie dann als der Schuldige gemacht.

Es ist ganz offensichtlich, dass Sie ein paar Schutzvorkehrungen treffen müssen, wenn Anwender – und besonders Neulinge – Ihre Arbeitsmappen verwenden. Excel stellt verschiedene Techniken zur Verfügung, um Arbeitsblätter und Teile davon zu schützen:

- ✓ **Bestimmte Zellen sperren:** Sie können bestimmte Zellen sperren und so Änderungen an den Zellen verhindern. (Verwenden Sie hierzu die Registerkarte SCHUTZ des Dialogfelds ZELLEN FORMATIEREN.) Die Sperrung ist dann aktiv, wenn das Dokument mit dem Befehl ÜBERPRÜFEN|ÄNDERUNGEN|BLATT SCHÜTZEN geschützt wurde. Im Dialogfeld BLATT SCHÜTZEN finden Sie verschiedene Optionen, mit denen Sie festlegen können, welche Aktionen die Anwender in einem geschützten Blatt vornehmen können.

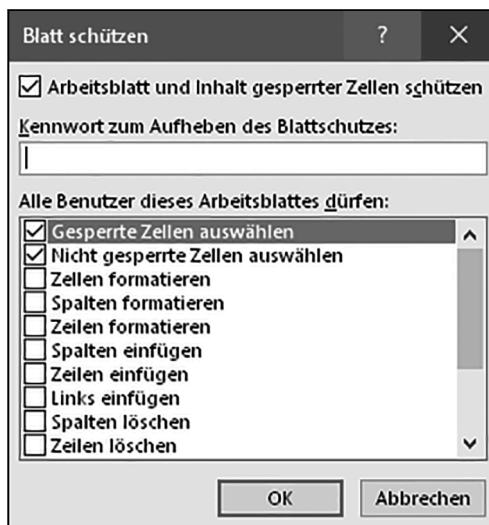


Abbildung 1.5: Verwenden Sie das Dialogfeld *BLATT SCHÜTZEN*, um festzulegen, was Benutzer dürfen und was nicht.

- ✓ **Formeln in bestimmten Zellen ausblenden:** Sie können Formeln in bestimmten Zellen ausblenden, damit andere sie nicht sehen können. (Verwenden Sie hierzu die Registerkarte SCHUTZ des Dialogfelds ZELLEN FORMATIEREN.) Auch hier wird dieser Schutz nur dann aktiv, wenn das Dokument mit dem Befehl ÜBERPRÜFEN|ÄNDERUNGEN|BLATT SCHÜTZEN geschützt wurde.
- ✓ **Gesamte Arbeitsmappe schützen:** Sie können die gesamte Arbeitsmappe schützen – die Struktur der Arbeitsmappe, die Position und Größe der Fenster, oder beides. Verwenden Sie für diesen Zweck den Befehl ÜBERPRÜFEN|ÄNDERUNGEN|ARBEITSMAPPE SCHÜTZEN.
- ✓ **Objekte auf dem Arbeitsblatt sperren:** Verwenden Sie den Abschnitt EIGENSCHAFTEN im Aufgabenbereich, um Objekte (wie beispielsweise Formen) zu sperren und zu verhindern, dass sie geändert oder verschoben werden. Um diesen Abschnitt zu öffnen, klicken Sie das

Objekt mit der rechten Maustaste an und wählen Sie GRÖSSE UND EIGENSCHAFTEN. Das Sperren der Objekte ist nur dann aktiv, wenn Sie das Dokument mit dem Befehl ÜBERPRÜFEN|ÄNDERUNGEN|BLATT SCHÜTZEN schützen. Standardmäßig sind alle Objekte gesperrt.

- ✓ **Zeilen, Spalten, Blätter und Arbeitsmappen verbergen:** Sie können ganze Zeilen, Spalten, Blätter und Arbeitsmappen ausblenden. Hierdurch wirkt das Arbeitsblatt weniger überladen; gleichzeitig bietet dies einen einfachen Schutz vor neugierigen Blicken.
- ✓ **Schreibschutz für eine Arbeitsmappe empfehlen:** Sie können den Schreibschutz für eine Arbeitsmappe empfehlen (und ein Kennwort vergeben), damit die Datei nicht mit einer geänderten Version überschrieben wird. Sie nehmen diese Einstellungen im Dialogfeld ALLGEMEINE OPTIONEN vor. Das Dialogfeld wird geöffnet, indem Sie DATEI|SPEICHERN UNTER wählen, die Schaltfläche Tools anklicken und dann den Befehl ALLGEMEINE OPTIONEN wählen.
- ✓ **Kennwort vergeben:** Sie können eine Arbeitsmappe mit einem Kennwort versehen und so verhindern, dass nicht autorisierte Personen Ihre Datei öffnen. Verwenden Sie hierfür den Befehl DATEI|INFORMATIONEN|ARBEITSMAPPE SCHÜTZEN|MIT KENNWORT VERSCHLÜSSELN.
- ✓ **Kennwortgeschütztes Add-In verwenden:** Sie können ein kennwortgeschütztes Add-In verwenden, wodurch die Benutzer überhaupt keine Änderungen auf den Arbeitsblättern des Add-Ins vornehmen können.

### ***Excel-Kennwörter bieten keinen 100 %igen Schutz***

Bitte bedenken Sie, dass die Excel-Kennwörter einfach geknackt werden können, indem man kommerzielle Tools verwendet. Zwar wurde ab Excel 2007 die Sicherheit im Vergleich zu den Vorgängerversionen verbessert, jedoch kann ein entschlossener Benutzer sie noch immer knacken. Die Moral: Gehen Sie nicht davon aus, dass der Kennwortschutz absolut sicher ist. Zugegeben, für den durchschnittlichen Benutzer reicht diese Form der Sicherheit aus. Fall jemand wirklich Ihr Kennwort knacken will, wird ihm dies vermutlich auch gelingen.

### ***Erstellen Sie eine ästhetisch ansprechende und intuitive Anwendung***

Wenn Sie bereits mehrere Softwareanwendungen verwendet haben, dann kennen Sie zweifellos Beispiele von schlecht entworfenen Benutzeroberflächen, schwer zu bedienenden Programmen und einfach nur hässlichen Bildschirmen. Falls Sie Tabellenkalkulationen für andere Leute entwickeln, dann sollen Sie ein besonderes Augenmerk darauf legen, wie die Anwendung aussieht.

Das Aussehen eines Computerprogramms kann für die Anwender den Unterschied ausmachen; das Gleiche gilt auch für Anwendungen, die Sie mit Excel entwickeln. Jedoch liegt die Schönheit im Auge des Betrachters. Falls Ihre Fähigkeiten eher im analytischen Bereich liegen, sollten Sie in Erwägung ziehen, sich bei gestalterischen Fragen von jemandem mit einem Gespür für Ästhetik helfen zu lassen.

Die gute Nachricht ist, dass sich mit den Features in Excel 2007 und neuer einfacher gut aussehende Tabellenkalkulationen entwickeln lassen. Wenn Sie vor allem die vordefinierten Zellenformatvorlagen verwenden, ist die Wahrscheinlichkeit groß, dass Ihre Arbeit gut aussieht. Außerdem können Sie dann mit ein paar Mausklicks ein neues Design zuweisen, das das Aussehen der gesamten Arbeitsmappe ändert – und immer noch gut aussieht.

Endbenutzer schätzen eine gut aussehende Benutzeroberfläche. Wenn Sie ein wenig Zeit mit Fragen des Designs und mit ästhetischen Überlegungen verbringen, macht Ihre Anwendung einen ausgereifteren und professionelleren Eindruck. Eine Anwendung, die gut aussieht, vermittelt den Eindruck, dass der Entwickler die Anwendung so wichtig nimmt, dass er bereit ist, die zusätzliche Zeit und den zusätzlichen Aufwand in Kauf zu nehmen. Berücksichtigen Sie die folgenden Empfehlungen:

- ✓ **Streben Sie Konsistenz an.** Wenn Sie beispielsweise Dialogfelder verwenden, sollten Sie, wann immer möglich, die Anmutung der Excel-Dialogfelder emulieren. Seien Sie bei der Formatierung, den Schriftarten, der Textgröße und den Farben konsistent.
- ✓ **Machen Sie es nicht zu kompliziert.** Entwickler begehen häufig den Fehler, zu viele Informationen in einen Bildschirm oder in ein Dialogfeld zu packen. Eine gute Faustregel ist, gleichzeitig immer nur ein oder zwei Informationsblöcke zu präsentieren.
- ✓ **Teilen Sie Eingabemasken auf.** Wenn Sie eine Eingabemaske verwenden, um so beim Anwender Informationen abzufragen, sollten Sie erwägen, große Masken in mehrere, weniger überladene Masken aufzuteilen. Falls Sie ein umfangreiches Dialogfeld verwenden, können Sie es mithilfe eines MultiPage-Steuerelements aufteilen und dabei die vertrauten Registerkartendialogfelder verwenden.
- ✓ **Übertreiben Sie den Einsatz von Farben nicht.** Verwenden Sie Farben sparsam. Es kann sehr leicht passieren, dass man es bei der Verwendung von Farben übertreibt und die Anwendung zu knallig wirkt.
- ✓ **Behalten Sie die Typografie und die grafische Gestaltung im Auge.** Achten Sie auf die numerischen Formate und verwenden Sie konsistente Schriftarten, Schrittgrößen und Rahmen.

Die Bewertung von ästhetischen Merkmalen ist subjektiv. Falls Sie sich nicht ganz sicher sind, sollten Sie immer Einfachheit und Klarheit anstreben.

### ***Ein Hilfesystem erstellen***

Bei der Erstellung der Anwenderdokumentation haben Sie im Prinzip zwei Optionen. Sie können die Dokumentation auf Papier oder elektronisch erstellen. Bei Windows-Anwendungen gehört eine elektronische Dokumentation zur Standardausstattung. Glücklicherweise können Sie auch in Excel-Anwendungen Hilfe bereitstellen; auch kontextsensitive Hilfe ist machbar. Die Entwicklung eines Hilfesystems benötigt zusätzliche Zeit. Jedoch kann sich diese Mühe insbesondere bei größeren Projekten lohnen.

Weiterhin sollten Sie überlegen, wie Sie Support für Ihre Anwendung anbieten. Wer kann beispielsweise angerufen werden, wenn Anwender auf ein Problem stoßen? Falls Sie nicht in der Lage sind, Routinefragen zu beantworten, dann sollten Sie jemanden bestimmen, der sich

darum kümmern kann. In bestimmten Fällen kann es sinnvoll sein, dass lediglich technisch komplexe oder Fragen, die sich auf Bugs beziehen, an den beziehungsweise die Entwickler weitergeleitet werden.



In Kapitel 19 stellen wir verschiedene Alternativen für das Bereitstellen von Hilfe für Ihre Anwendungen vor.

## ***Dokumentieren Sie Ihre Entwicklungsarbeit***

Eine Tabellenkalkulationsanwendung zu erstellen ist eine Sache. Sie auch für andere Leute verständlich und nachvollziehbar zu machen, ist eine andere. Wie bei der traditionellen Programmierung ist es wichtig, Ihre Arbeit gut zu dokumentieren. Diese Dokumentation hilft Ihnen, wenn Sie zu einem späteren Zeitpunkt beispielsweise Änderungen vornehmen wollen. Sie hilft auch anderen, die die Weiterentwicklung übernehmen.

Wie dokumentieren Sie eine Arbeitsblattanwendung? Sie können diese Informationen entweder in ein Arbeitsblatt einfügen oder hierfür eine eigene Datei verwenden. Sie können die Dokumentation auch auf Papier erstellen, falls Sie dies bevorzugen. Der einfachste Weg ist vielleicht, ein eigenes Arbeitsblatt für Ihre Notizen und Kommentare zu verwenden, und dort auch die entscheidenden Informationen über das Projekt zu speichern. Verwenden Sie in VBA-Code so viele Kommentare, wie Sie möchten. (Text, der in VBA mit einem Hochkomma eingeleitet wird, wird als Kommentar behandelt.) Auch wenn ein elegantes Stück VBA-Code heute vollkommen verständlich ist, so kann der Code in ein paar Monaten, wenn Sie ihn beispielsweise anpassen müssen, viele Fragen aufwerfen – es sei denn, Sie haben den Code gut mit dem VBA-Kommentarfeature dokumentiert.

## ***Die Anwendung an die Benutzer verteilen***

Sie haben Ihr Projekt fertiggestellt und können es an die Endbenutzer verteilen. Wie gehen Sie hierbei vor? Sie können für die Verteilung Ihrer Anwendung verschiedene Wege einschlagen. Welche Methode Sie verwenden, hängt von vielen Faktoren ab.

Sie können beispielsweise eine CD-ROM oder einen USB-Stick erstellen, ein paar Anweisungen skizzieren und sind fertig. Oder Sie nehmen die Installation selbst vor Ort vor, jedoch ist dies nicht immer ein gangbarer Weg. Eine andere Option besteht darin, ein offizielles Installationsprogramm zu entwickeln, das diese Aufgabe automatisch erledigt. Sie können solch ein Programm in einer traditionellen Programmiersprache erstellen, ein Tool für die Erstellung von Setup-Programmen erwerben oder Ihr eigenes Setup-Programm in VBA programmieren.

Excel stellt die Technologien bereit, mit denen Entwickler ihre Anwendungen digital signieren können. Hierdurch können Endbenutzer prüfen, wer der Autor der Anwendung ist und ob das Projekt zwischenzeitlich geändert wurde. Dies hilft dabei, der Verbreitung von Makroviren und anderem, potenziell gefährlichem Code vorzubeugen. Um ein Projekt digital zu signieren, erwerben Sie zuerst ein digitales Zertifikat von einer autorisierten Zertifizierungsstelle (oder Sie verwenden die Office-Tools, um ein selbst signiertes Zertifikat zu erstellen). Im Hilfesystem und auf der Microsoft-Website finden Sie weitere Informationen zu diesem Thema.



### ***Die Anwendung updaten, falls nötig***

Nachdem Sie die Anwendung verteilt haben, sind Sie damit fertig, richtig? Sie können sich zurücklehnen, sich an der getanen Arbeit erfreuen, und alle Probleme vergessen, auf die Sie während der Entwicklungsphase gestoßen sind und die Sie gelöst haben. In ganz wenigen Fällen wird es auch so sein, dass Sie fertig sind. Viel öfter wird es vorkommen, dass die Benutzer mit Ihrer Anwendung nicht komplett zufrieden sind. Zugegeben, Ihre Anwendung erfüllt die *ursprünglichen* Spezifikationen, jedoch ändern sich die Dinge. Ist eine Anwendung erst mal beim Benutzer im Einsatz, wird er viele neue Sachen entdecken, die die Anwendung ebenfalls können sollte.

Wenn Sie Ihre Anwendung aktualisieren oder anpassen müssen, werden Sie vor allem froh sein, dass sie ein gutes Design aufweist und dass Sie die internen Vorgänge sauber und umfassend dokumentiert haben.

### ***Weitere Punkte, die für die Entwicklung wichtig sind***

Sie müssen ein paar weitere Aspekte bedenken, wenn Sie eine Anwendung entwickeln – und zwar besonders dann, wenn Sie nicht ganz genau wissen, wer Ihre Anwendung einsetzen wird. Wenn Sie eine Anwendung entwickeln, die weit verbreitet eingesetzt wird (wie beispielsweise eine Shareware-Anwendung), dann wissen Sie nicht, wie Ihre Anwendung eingesetzt wird, auf welchem System sie benutzt wird und welche andere Software gleichzeitig läuft.

### ***Die beim Anwender installierte Excel-Version***

Trotz der Verfügbarkeit von Excel 2016 setzen viele große Unternehmen immer noch ältere Versionen von Excel ein. Leider gibt es keine Garantie dafür, dass eine Anwendung, die beispielsweise für Excel 2010 entwickelt wurde, auch mit neueren Versionen von Excel einwandfrei funktioniert. Wenn Ihre Anwendung mit unterschiedlichen Excel-Versionen funktionieren soll, ist der beste Ansatz, sie mit der ältesten benötigten Version zu entwickeln und die Anwendung anschließend gründlich mit allen anderen Versionen zu testen.

Achten Sie auch auf Servicepacks und Sicherheitsupdates, die von Microsoft veröffentlicht werden. Auch wenn es selten vorkommt, so kann es doch passieren, dass in diesen Releases vorgenommene Änderungen dazu führen, dass bestimmte Komponenten Ihrer Anwendung nicht mehr wie beabsichtigt funktionieren.



Kompatibilitätsaspekte werden in Kapitel 21 erläutert.

### ***Probleme aufgrund der Sprache***

Schätzen Sie sich glücklich, wenn all Ihre Endbenutzer die englischsprachige Version von Excel verwenden. Die nicht englischen Versionen sind nicht immer zu 100 Prozent kompatibel und Sie müssen daher zusätzlichen Testaufwand betreiben. Bedenken Sie, dass zwei Anwender zwar die gleiche Sprachversion von Excel und dennoch in den Systemeinstellungen

andere Optionen für Region und Sprache verwenden können. In bestimmten Fällen müssen Sie auf Probleme achten, die sich hieraus ergeben können.



Probleme im Zusammenhang mit der Sprache werden in Kapitel 21 kurz angerissen.

## **Systemgeschwindigkeit**

Vermutlich sind Sie ein ziemlich fortgeschrittener Computeranwender und halten Ihre Hardware so gut es geht auf dem aktuellen Stand. Daher setzen Sie vermutlich ein System ein, das leistungstärker ist als die Hardware des durchschnittlichen Benutzers Ihrer Anwendung. In manchen Fällen kennen Sie die Hardware- und Software-Umgebung, auf der Ihre Anwendung ausgeführt wird, sogar ziemlich genau. Sollte dies der Fall sein, ist es wichtig, Ihre Anwendung auf genau solch einem System zu testen. Eine Prozedur, die auf Ihrem System in Sekundenbruchteilen ausgeführt wird, mag auf einem anderen System mehrere Sekunden benötigen. In der Computerwelt sind Wartezeiten von mehreren Sekunden nicht akzeptabel.



Wenn Sie mehr Erfahrungen bei der Programmierung in VBA gesammelt haben, werden Sie entdecken, dass es Wege gibt, um eine Aufgabe zu erledigen, und Wege, um diese schnell zu erledigen. Es ist ein guter Ansatz, beim Schreiben von Code immer auch die Ausführungsgeschwindigkeit im Auge zu behalten. Tipps und Tricks in anderen Kapiteln dieses Buches unterstützen Sie dabei.

## **Videomodi**

Wie Sie sicherlich wissen, gibt es erhebliche Unterschiede bei den Monitoren. Die derzeit am weitesten verbreiteten Bildschirmauflösungen sind  $1366 \times 768$ ,  $1280 \times 800$  und  $1920 \times 1080$  Pixel. Systeme mit einer Auflösung von  $800 \times 600$  Bildpunkten sind heutzutage weniger verbreitet, werden aber immer noch verwendet. Monitore mit höherer Auflösung und auch Systeme mit zwei Bildschirmen erfreuen sich immer größerer Beliebtheit. Nur weil Sie einen Monitor mit einer sehr hohen Auflösung verwenden, können Sie nicht davon ausgehen, dass auch alle anderen dies tun.

Die Bildschirmauflösung kann ein Problem darstellen, wenn sich Ihre Anwendung darauf verlässt, dass bestimmte Informationen auf einer Bildschirmseite dargestellt werden können. Wenn Sie beispielsweise eine Eingabemaske für eine Auflösung von  $1280 \times 1024$  erstellen, bei der alle Informationen gleichzeitig sichtbar sind, werden Anwender, die eine Auflösung von  $1024 \times 768$  Pixel verwenden, nur mit Scrollen oder Zoomen in der Lage sein, alle Eingabefelder zu sehen.

Außerdem sollten Sie bedenken, dass eine Arbeitsmappe, deren Größe *wiederhergestellt* wurde (sie wird also weder minimiert noch maximiert dargestellt), in ihrer ursprünglichen Größe und an ihrer ursprünglichen Position angezeigt wird. In extremen Fällen kann es passieren, dass ein Fenster, das bei einer sehr hohen Auflösung gespeichert wurde, vollkommen außerhalb des Bildschirms steht, wenn es auf einem System mit einer niedrigeren Auflösung geöffnet wird.

Leider ist es nicht möglich, die Dinge automatisch so zu skalieren, dass sie unabhängig von der Bildschirmauflösung immer gleich aussehen. In bestimmten Fällen können Sie ein Arbeitsblatt zoomen (indem Sie den Zoom-Regler in der Statusleiste verwenden), dies zuverlässig zu tun, ist jedoch schwierig. Bis auf die Fälle, in denen Sie absolut sicher sind, welche Bildschirmauflösung die Benutzer Ihrer Anwendung verwenden, sollten Sie Ihre Anwendung so gestalten, dass sie beim kleinsten gemeinsamen Nenner gut funktioniert – dies könnte beispielsweise eine Auflösung von  $800 \times 600$  oder  $1024 \times 768$  Bildpunkten sein.

Wie Sie im weiteren Verlauf des Buches noch sehen werden, können Sie von VBA aus Windows-API-Funktionen aufrufen, um die aktuelle Bildschirmauflösung des Anwenders zu ermitteln. In bestimmten Fällen können Sie dann die Benutzeroberfläche vom Code aus justieren, damit sie besser zur vom Anwender eingestellten Bildschirmauflösung passt.

