

Notizen Office 1

Klaus Kusche, 2012 / 2013

1. Kapitel: Office-Anwendungen

Anwendungen

Welche Anwendungen gehören zu einem Office-Paket?

- *Textverarbeitung* (“MS Word”, “OO Writer”):
Texte und Dokumente aller Art
Serienbriefe
- *Tabellenkalkulation* (“MS Excel”, “OO Calc”):
Berechnungen in Tabellenform
Einfache Datenbanken
Daten-Auswertung (Diagramme und Grafiken)
- *Präsentationsprogramm* (“MS PowerPoint”, “OO Impress”):
Präsentationen (auch animiert), Folien, Vortragsunterlagen
- *“Datenbank” (?)* (“MS Access”, “OO Base”):
Datenhaltung in Tabellen-Format
Suche und Auswertungen
- Für Informatiker oft wichtig:
Diagramm-Zeichner (“MS Visio”, “Dia”, “KDE Kivio”,
nur eingeschränkt “OO Draw”, “Inkscape”)

Meist separat:

- *Desktop Publishing* (“Adobe InDesign”, “QuarkXPress”, “MS Publisher”, “Scribus”):
Gestaltung von Zeitschriften, Flyern, Plakaten, ...

File-Formate

Unterscheide:

- Internes Format: .doc / .docx, .odt, ...
- Exportierte Formate: Wichtig vor allem: .pdf, .html

Verwendung:

- Zur Weiterverarbeitung: Immer im internen Format speichern!
- Zur Weitergabe, Archivierung, Veröffentlichung: In Standard-Format exportieren!

Interne Formate:

Vorteile:

- Nur interne Formate bieten alle Möglichkeiten der Software und erhalten die volle

Formatierung!

(bei Export und Import gehen meist Formatierungen und Funktionen verloren!)

Nachteile:

- Interne Formate sind hersteller-spezifisch / plattform-spezifisch und mit Fremdprogrammen nicht / schlecht lesbar.
(Word-Dokument am Handy? PowerPoint-Präsentation unter Linux?)
Beim Einsatz in Firmen: Hersteller-Bindung, beträchtlicher Umstellungsaufwand!
- Interne Formate veraltern schneller.
(Was ist mit dem Word-Dokument in 15 Jahren?)
- Interne Formate enthalten keine Fonts / verweisen auf Systemfonts
=> hat der Empfänger dieselben Fonts installiert?
- Interne Formate sind binär und nicht / schlecht genormt:
 - Tatsächlicher Inhalt nicht kontrollierbar (oft: Alte Versionen, ausgeblendete Daten, Reste anderer Dokumente, Cut-and-Paste-History, ...)
=> Großes Risiko von Daten-Leaks bei Versand via Mail oder Veröffentlichung am Internet!
 - Von Fremd-Such-Software und Fremd-Index-Erstellern nur schlecht erfassbar
=> Information in Word / OO-Dokumenten schlecht auffindbar!
 - Nicht sinnvoll Versions-verwaltbar (keine zeilenweisen Unterschiede, keine effiziente Speicherung nur der Änderungen)
=> **Für Dokumentation professioneller Software nur schlecht geeignet!**
 - Reparatur "abgestürzter" oder teilweise beschädigter / verlorener Dokumente schwierig: Auch unbeschädigte Teile sind nicht mehr zu öffnen.
- Interne Formate sind ein größeres Sicherheits-Risiko.
 - Sicherheitslücken, Viren usw.
 - Makros

Daher bei Firmen / Behörden oft:

An Firewalls gefiltert, Makros generell ausgeschaltet, ...

- Bei Veröffentlichung: Interne Formate sind relativ leicht modifizierbar / fälschbar.
- Nachbildung strukturierter, verlinkter Information ist schwierig
=> Ungeeignet für interaktive Hilfe / Dokumentation "zum Durchklicken".
- MS-spezifisches Problem, zumindest in alten Office-Versionen:
Formatierung (Zeilenumbruch / Seitenumbruch) ist abhängig vom Druckertreiber!
(bei Wechsel des Druckers / Update der Fonts "fällt das Dokument auseinander").

Alternative .pdf:

Verwendung für **Versand / Veröffentlichung** von Dokumenten,

- bei denen es auf exakten Erhalt des Aussehens ankommt,
- oder die normalerweise gedruckt oder sequentiell gelesen werden.

Verwendung für **Archivierung** sich nicht mehr ändernder Dokumente.

Vorteil:

- **Absolut formatierungs- und Layout-erhaltend**
==> schaut immer und überall gleich aus.
- Fast überall lesbar, mehrere Software-Alternativen zum Lesen.
- Genormt (zumindest alte Versionen), gesetzlich zulässiges Archiv-Format, wird nicht so schnell veralten.
- Bei Fonts: Wahlweise extern oder eingebettet ("eingebettet" angeben!)
==> Der Leser sieht denselben Font.
- Schwerer nachträglich zu verfälschen.

Nachteile:

- Nicht mehr nachträglich bearbeitbar:
Inhalt ist "abfotografiert", logische Struktur praktisch verloren.
- Ebenfalls binär.
- Verlinkte Strukturierung auch mühsam.
- In letzter Zeit zahlreiche Sicherheits-Probleme...

Hinweise zu .pdf:

- PDF-Erzeugung: Bei OpenOffice/LibreOffice direkt (Qualität sehr gut!), bei altem MS Office entweder über "PDF-Drucker" oder über Webdienste.
- **PDF/A** verwenden: Genormtes Format, zur gesetzeskonformen Archivierung zugelassen, auch mit alternativer PDF-Software verwendbar (zur Not: Alte .pdf-Standards, <= 1.4).
- Wenn bei der Erzeugung die Wahl angeboten wird: Fonts embedden!

Alternative .html:

Verwendung für

- **strukturierte Information** zum Durchklicken (Hilfe / SW-Doku)
- **und Veröffentlichung am Internet.**

.html beschreibt die Information und die Struktur, aber **nicht das Layout!**

Vorteile:

- Passt sich dem Ausgabegerät an (Bildschirmgröße, Handy).
- Passt den Seiteninhalt der Font-Größe an (Leser mit Seh-Schwäche).
- Standardisiert, portabel, auf fast jedem Rechner ohne eigene SW und "ewig" lesbar.
- Strukturierte Informations-Darstellung, schnelle Navigation durch "Durchklicken".
- Wesentliche Textstruktur bleibt erhalten, zur Not auch als Bearbeitungsformat geeignet (aber dadurch auch jederzeit nachträglich zu fälschen).
- Reines Textformat:
Inhalt genau bekannt / prüfbar.

Mit externen Tools durchsuchbar / bearbeitbar / versionsverwaltbar / ...
Zur Not mit Notepad o.ä. reparierbar.
Garantiert ohne Sicherheitsgefahren!

Nachteile:

- Sieht bei fast jedem Leser anders aus...
(vorhandene und eingestellte Fonts, Fenstergröße, Browser-Unterschiede)
- Formatierung sehr simpel, komplexe Formatierungen gehen verloren.
- Seiten-Aufteilung geht verloren, Inhaltsverzeichnis damit wenig hilfreich.
- Keine eingebetteten Fonts, auf installierte Fonts beim Leser angewiesen.
- Ausdruck problematisch:
Seitlich abgeschnittene Zeilen / Grafiken, willkürlicher Seitenumbruch, ...
- Keine Vektor-Grafik, Grafik skaliert weder mit Seitengröße noch mit Zoom bzw. Verstellung der Font-Größe mit.
- Strukturierte Dokumente, Dokumente mit Grafiken usw. bestehen aus vielen Files
==> Kopieren / Archivieren, Versand per Mail usw. aufwändiger!

Grafik-Formate:

Unterscheide:

- **Pixelgrafik** (verlustbehaftet komprimiert): .jpg
- **Pixelgrafik** (verlustfrei komprimiert): .gif, .tiff, .png
- **Vektorgrafik**: .svg, (.pdf)
(speichert Bildelemente wie Kreis, Linie, ... auflösungsunabhängig)

Verwendung:

- Für *Fotos*: .jpg
- Für *Screenshots* von Dialogen, *gescannte schwarzweiß-Texte*, ...: .gif / .tiff
- Für Diagramme, Grafiken, Logos, Icons: .svg oder ähnliches
(weil rasterfrei skalierbar, auch hochauflösend scharf!)

Hinweis zu Fonts:

Auch Fonts unterliegen einem Copyright!

Bei Veröffentlichung von .pdf mit embedded Fonts, Upload von HTML mit eigenen Fonts:
Zuerst Copyright prüfen, sonst Schadenersatzforderungen / Unterlassungserklärungen!

Alternativen zu Textverarbeitung / Präsentationsprogramm

1.) Desktop-Publishing-SW:

Bei aufwändig grafisch gestalteten Seiten / Schrifteffekten.

2.) HTML-Editoren:

Theoretisch eine gute Idee, aber...

- Freie HTML-Editoren (z.B. Seamonkey oder Kompozer) können deutlich weniger als OO Writer im HTML-Mode ==> Gleich OO Writer verwenden?!

Achtung:

OO-Writer gleich beim Anlegen des Dokumentes im HTML-Mode starten ("OO Writer/Web"): Mit "Datei->Neu->HTML-Dokument"

Dann kommt "schönes" (menschlich les- und bearbeitbares) HTML heraus.

OO-Writer im normalen Modus mit HTML-Export produziert hingegen fast ebenso überladenes und unverständliches HTML wie MS Word...

- Gute professionelle HTML-Editoren kosten sehr viel...

3.) TeX / LaTeX:

(<http://www.latex-project.org/> <http://www.gierhardt.de/TeX/> <http://www.dante.de/>)

- Freie Software, ursprünglich von Prof. Knuth (Stanford), einem der "Väter" der Informatik.
- Erzeugt aus normalen Textfiles mit händisch eingetippten Formatierbefehlen (Struktur ähnlich HTML) nachträglich .pdf-Dokumente.
- Ziel ist nicht Benutzer-Komfort, sondern professioneller Textsatz, d.h. Buchdruck. Typografisch sehr hohe Qualität (wesentlich besser als MS Word / OO Writer), vor allem bei math. Formeln. Hohe Stabilität (auch bei tausenden Seiten).
- Seit über 30 Jahren Quasi-Standard für Bücher und wissenschaftliche Arbeiten der Mathematik / Informatik / Naturwissenschaften, nach wie vor an einigen Uni's für Diplomarbeiten / Dissertationen vorgeschrieben. Auch für gedruckte SW-Dokumentation verwendet.
- "What-you-see-is-what-you-get"-Editoren verfügbar (LyX).

4.) AsciiDoc: (<http://www.methods.co.nz/asciidoc/>)

Macht aus "natürlich" aufbereiteten Textfiles ("=" als Unterstreichung für Überschriften, "-" am Beginn von Aufzählungs-Elementen, ...) nachträglich .pdf-Dokumente oder .html-Seiten. Auch Open Source.

5.) SW-Dokumentations-Generatoren:

Erzeugen aus speziell markierten Kommentaren im Quelltext automatisch strukturierte .html-Hilfe oder gedruckte .pdf-Dokumentation.

Beispiele:

Doxygen für C / C++ / ... (<http://www.doxygen.org/>)

Javadoc für Java (<http://de.wikipedia.org/wiki/Javadoc>)