

Emacs Org-mode

der hackbare Notizblock in Plaintext

Florian Adamsky

`florian-gpn12@adamsky.it`
`http://florian.adamsky.it`

Gulaschprogrammierenacht 2012



Outline

- 1 Einleitung
- 2 Dokumenten Struktur
- 3 Aufgaben
- 4 Erweiterte Struktur
- 5 Arbeit mit Quellcode
- 6 Publizieren
- 7 Ausblick



Contents

- 1 Einleitung
- 2 Dokumenten Struktur
- 3 Aufgaben
- 4 Erweiterte Struktur
- 5 Arbeit mit Quellcode
- 6 Publizieren
- 7 Ausblick



Was ist Org-Mode

- ein Major-Mode für Emacs:
 - Notizen schreiben
 - Todo-Listen anlegen und verwalten
 - Projekte planen
 - und vieles mehr
- vereint Notizen mit Aufgaben
- Org-mode verwendet Plaintext-Dateien
 - portabel
 - einfach
 - zukunftssicher
 - VCS
- natürlich frei
 - GNU GPL 3



Geschichte

- entwickelt von Carsten Dominik
 - aktueller Maintainer: Bastien Guerry
- Org-mode baut auf outline-mode auf
 - grausame Tastenkürzel um die Gliederung zu ändern
 - 2003: “outline-magic.el”
- Seit Emacs 22.1 ist Org-mode fester Bestandteil



Aktivierung

- Alle Dateien mit der Endung *.org öffnet Emacs mit Org-mode:

```
1 (add-to-list 'auto-mode-alist '("\\.org\\")  
2                               . org-mode))
```

- Empfohlene Tastenkürzel:

```
1 (global-set-key "\C-cl" 'org-store-link)  
2 (global-set-key "\C-cc" 'org-capture)  
3 (global-set-key "\C-ca" 'org-agenda)  
4 (global-set-key "\C-cb" 'org-iswitchb))
```



Contents

- 1 Einleitung
- 2 Dokumenten Struktur**
- 3 Aufgaben
- 4 Erweiterte Struktur
- 5 Arbeit mit Quellcode
- 6 Publizieren
- 7 Ausblick



Überschriften

- Überschriften definieren die Struktur eines Dokuments
 - beginnen mit einem Stern: *
- Reihenfolge ändern: **Alt** + **↑** **↓**
- Hierarchie ändern: **Alt** + **←** **→**

```

* Überschrift
** zweite Überschrift
*** dritte Überschrift
**** vierte Überschrift
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec hendrerit tempor
tellus. Donec pretium posuere tellus. Proin quam nisl, tincidunt et, mattis eget,
convallis nec, purus. Cum sociis natoque penatibus et magnis dis parturient
montes, nascetur ridiculus mus. Nulla posuere. Donec vitae dolor. Nullam tristique
diam non turpis. Cras placerat accumsan nulla. Nullam rutrum. Nam vestibulum
accumsan nisl.
* Weitere Überschrift
** Foofoo
** Barfoo
*** Foofoo Barfoo
** Test-Überschrift
*** dritte Überschrift
**** vierte Überschrift
***** fünfte Überschrift
***** sechste Überschrift

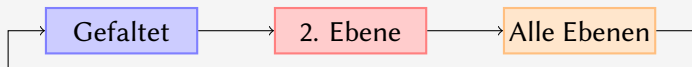
```





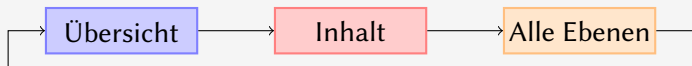
Sichtbarkeit

- Sichtbarkeit der Struktur ändern mit:




- Org-Cycle: 



- Org-Global-Cycle:  + 



Listen

- Ungeordnete Listen: -, + und *
- Geordnete Listen: 1. oder 1)
- Beschreibung: Test :: Beschreibung
- ebenfalls alles faltbar
-  +   iteriert durch die Listentypen

** Lord of the Rings

My favorite scenes are (in this order)

1. The attack of the Rohirrim
2. Eowyn's fight with the witch king
+ this was already my favorite scene in the book
+ I really like Miranda Otto.
3. Peter Jackson being shot by Legolas
- on DVD only

He makes a really funny face when it happens.

But in the end, no individual scenes matter but the film as a whole.

Important actors in this film are:

- Elijah Wood :: He plays Frodo
- Sean Austin :: He plays Sam, Frodo's friend. I still remember him very well from his role as Mikey Walsh in The Goonies.



Sparse Trees

- ermöglichen eine selektive Ansicht des Dokuments
- Tastenkürzel: `Strg+c` `/`
- oft verwendete Ansichten können gespeichert werden



Contents

- 1 Einleitung
- 2 Dokumenten Struktur
- 3 Aufgaben**
- 4 Erweiterte Struktur
- 5 Arbeit mit Quellcode
- 6 Publizieren
- 7 Ausblick



Aufgaben

- Überschriften werden zu einer Aufgabe wenn diese mit **TODO** beginnen
- *** TODO Vortrag für die GPN12 vorbereiten
- Standardmäßig gibt es zwei Zustände: Todo und Done
- Status wechseln mit `Strg+c` `Strg+t`
- Verknüpfung mit Sparse Trees: `Strg+c` `/`
- eigene Zustände können definiert werden mit `#+SEQ_TODO`
 - `#+SEQ_TODO: BUG KNOWCAUSE | FIXED IAMSICKOFIT`



Zeitstempel

- Termin setzen: `Strg` + `c` `Strg` + `s`
- Deadline setzen: `Strg` + `c` `Strg` + `d`
- normales Datum einfügen mit: `Strg` + `.`
- sehr effizientes Interface:

Syntax	Bedeutung
d	Tage
w	Wochen
m	Monate
y	Jahre
+4d	4 Tage beginnend von heute
+1w	1 Woche beginnend von heute
+2tue	2 Dienstag beginnend von heute



Aufgaben on Steroids

* TODO Miete bezahlen

DEADLINE <2012-05-01 Sat +1m -3d >

- Wiederholungs-Turnus _____ ↗
- Warnungs-Periode _____ ↗
- Prioritäten setzen ⬆ + ⬆ ⬆
- Column-View mit Strg + c Strg + x Strg + c



Checkboxen und Teilaufgaben

- Checkbox startet mit []
 - `[Strg]+[c]` `[Strg]+[c]` markiert die Checkbox
- ähnliche wie Aufgaben
 - nicht in der globalen Liste verfügbar
- eignet sich gut um größere Aufgaben aufzuteilen
- Übersicht über die Teilaufgaben mit [/] oder [%]



Agenda-View

- globale Ansicht über alle Aufgaben
 - auch mehrere Dateien möglich (`Strg`+`c` `l` fügt aktuelle Datei hinzu)
- erreichbar über `Strg`+`c` `a`
- die folgenden Ansichten sind möglich:
 - Agenda-View (Tag, Woche, Monat und Jahr)
 - reine Aufgabenlisten
 - Tags
 - eigene Sparse Trees

```

Week-agenda (W23):
Monday 4 June 2012 W23
  private: Deadline: TODO Urlaub für GPN12 beantragen
Tuesday 5 June 2012
  private: Scheduled: TODO halte Prokrastinationsvortrag
  private: Deadline: TODO verschiebe Prokrastinationsvortrag auf einen anderen Tag
Wednesday 6 June 2012
  private: Deadline: TODO Vortrag für die GPN12 vorbereiten [50%]
Thursday 7 June 2012
  private: (1/4): <2012-06-07 Do>--<2012-06-10 So> GPN12
Friday 8 June 2012
  private: (2/4): <2012-06-07 Do>--<2012-06-10 So> GPN12
Saturday 9 June 2012
  private: (3/4): <2012-06-07 Do>--<2012-06-10 So> GPN12
Sunday 10 June 2012
  private: (4/4): <2012-06-07 Do>--<2012-06-10 So> GPN12
  
```

```

U:%*- *Org Agenda* All (2,0) [(Org-Agenda Week Ddl Grid pair v1 Wrap)] 7:15 1.44
Rebuilding agenda buffer...done
  
```



Contents

- 1 Einleitung
- 2 Dokumenten Struktur
- 3 Aufgaben
- 4 Erweiterte Struktur**
- 5 Arbeit mit Quellcode
- 6 Publizieren
- 7 Ausblick



Hyperlinks

- Org erkennt URLs automatisch und macht diese klickbar
- Syntax: `[[link][beschreibung]]` oder `[[link]]`
- `Strg`+`C` `I` kopiert den Ort
- `Strg`+`C` `Strg`+`I` fügt einen neuen Link ein
- vielfältige Auswahl an Links:

Kategorie	Link
E-Mail:	Mailto, Wanderlust, Gnus, Rmail, Mhe, VM
Emacs:	BBDB, Info, Elisp, Docview, Unix man pages
Internet:	Http, Https, Ftp, IRC, Usenet, DOI
Intern:	File (relative, absolut), Bibtex, Shell



Markup

Markup	Syntax
Fett	<code>*bold*</code>
Unterstrichen	<code>_underlined_</code>
Durchgestrichen	<code>+strike-through+</code>
Verbatim	<code>~verbatim~</code>
Code	<code>=code=</code>
Kommentar	<code># no comment</code>

- Fußnoten: `Strg c` - `Strg x` - `f`



Tags

- jede Überschrift kann man mit mehreren Tags versehen
- mit dem Tastenkürzel `Strg+c` `Strg+q` setzt man ein Tag
- Tags unterstützen Vererbung
- Filterung durch Sparse Tree oder Agenda View

```
* Besprechung                                     :Arbeit:
** Zusammenfassung von Frank                     :Notizen:Chef:
*** TODO Folien vorbereiten                    :Präsentation:
```



Properties

- key-value Paar für eine Überschrift
- Jede Überschrift kann mehrere Properties haben
- Properties unterstützen ebenso Vererbung
- Suche über Sparse-Trees oder Agenda-View



Tabellen Magie












Figure: Bild von Gary Overacre



Tabellen 2

■ Tabellen-Editor

- Tabellen beginnen mit |
- |- und  zeichnet eine horizontale Linie
- Zeilen vertauschen:  +  
- Spalten vertauschen:  +  
- Sortieren:  + 

■ Tabellenkalkulation

- Backend ist der Emacs Taschenrechner calc
- für Tabellen existiert auch ein Minor Mode: orgtbl



Contents

- 1 Einleitung
- 2 Dokumenten Struktur
- 3 Aufgaben
- 4 Erweiterte Struktur
- 5 Arbeit mit Quellcode**
- 6 Publizieren
- 7 Ausblick



Code Blocks

- Org-mode unterstützt die Einbindung von Quelltext

```
#+BEGIN_SRC <language> <switches>  
<code>  
#+END_SRC
```

- `Strg` + `C` `'` öffnet den Quelltext im jeweiligen Major-Mode
- Standardmäßig nur Emacs Lisp unterstützt
- nachträglich können die folgenden Sprachen aktiviert werden:
 - C/C++, CSS, Graphviz, Gnuplot, Java, Javascript, Ledger, Lilypond, Mscgen, Octave, Oz, Plantuml, R, Sass, GNU Screen, SQL, Awk, Clojure, ditaa, Haskell, \LaTeX , Lisp, MATLAB, Objective Caml, Org-mode, Perl, Ruby, Python, Scheme, Shell und SQLite



Ausführen von Code Blocks

- Ausführen via `Strg` + `c` `Strg` + `c`
- Ergebnis wird in den Org mode buffer eingefügt
- Shell unterstützt auch entfernten Zugang mittels SSH



Literate Programming

Definition

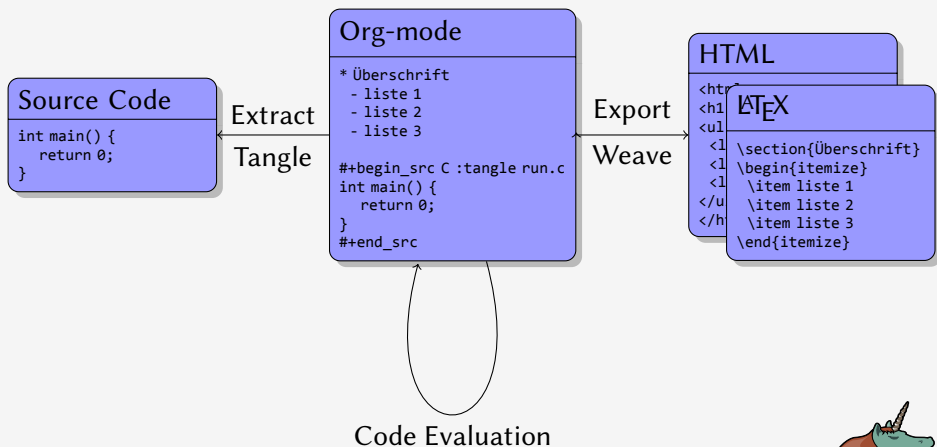
“Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.” - Donald Knuth (1980)

Tangle: extrahieren des Codes aus dem Dokument (computer-lesbar)

Weave: exportieren des Dokument in ein menschen-lesbares Format (HTML, \LaTeX)



Literate Programming mit Org-mode



Reproducible Research

Definition

“An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and complete set of instructions which generated the figures.” - Buckheit and Donoho (1995)



Contents

- 1 Einleitung
- 2 Dokumenten Struktur
- 3 Aufgaben
- 4 Erweiterte Struktur
- 5 Arbeit mit Quellcode
- 6 Publizieren**
- 7 Ausblick



Export

- Org mode unterstützt eine Vielzahl an Formaten zum Exportieren
 - HTML, \LaTeX , OpenDocument, ASCII, DocBook, Taskjuggler, Xoxo, iCal
 - `Strg+c` `Strg+e` ruft den Export-Dialog auf
- ganze Dokument oder nur Teile des Dokument
- selektiver Export mit Tags wie z.B.: `:noexport:`



Contents

- 1 Einleitung
- 2 Dokumenten Struktur
- 3 Aufgaben
- 4 Erweiterte Struktur
- 5 Arbeit mit Quellcode
- 6 Publizieren
- 7 Ausblick



Über was ich nicht gesprochen habe

- Capture
 - Notizen oder Aufgaben speichern ohne große Unterbrechungen
- Archive
 - Jede Überschrift lässt sich mit `Strg+c` `Strg+x` `Strg+a` automatisch in eine Archive-Datei überführen
- Zeit messen
- Hacking
 - Hooks überall



Integration mit Google Kalender

- Calfw - A calendar framework for Emacs
- <https://github.com/kiwanami/emacs-calfw>

2011 / January

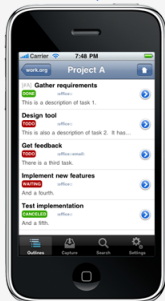
[[<]][>] [Today] [Day] [Week] [Two Weeks] [Month]

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
26	27	28	29	30	31	1 (1) 元日 TEST1
2 (1) PTEST1	3	4 New Moon	5	6	7	8 (1) PERIOD1-----
9 (1) PERIOD1-----	10 (4) 成... PTEST2 PTEST3 TEST2	11 (1) Period2-----	12 (2) Young (Long Long...)	13 (1) -----	14 (2) (Stack-----)	15 (1) -----
16	17	18	19	20 Full Moon	21	22
23	24	25	26 Waning Mo	27	28	29 (1) PERIOD W-----
30 (1) PERIOD W-----	31 (1) -----	1	2	3	4	5
[[test1]] [[test2]]						

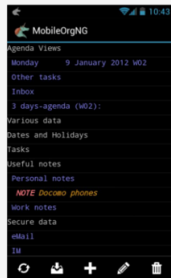


MobileOrg

iPhone/iPad



Android



Was kann man mit Org mode machen?

- Notizen schreiben
- Aufgabenliste
 - Getting Things Done ala David Allen
- Literate Programming
- Reproducible Research
- kleine Datenbank
- Framework um Webseiten zu erstellen
- Tool um Texte zu entwerfen (Papers, . . .)
- Taschenkettensäge für die Planung



Was Org mode nicht ist

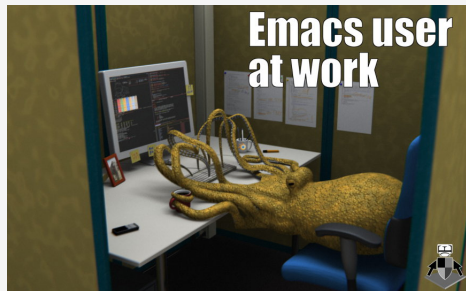
- frickel-frei
 - Emacs-user mögen frickeln
 - That's how I roll :-)
- gibt kein Patentrezept gegen Prokrastination
 - passt sich jedoch sehr flexibel den Gepflogenheiten an
- für Enduser
- vi benutzer ;-)



Zum Abschluß

“... there’s this awesome tool that gets overlooked, like people have some amazing sword in their cupboard and instead hack away at zombies with spoons.”

— Nic Ferrier



Weitere Informationen

Links

- <http://orgmode.org/>
- <http://www.gnu.org/software/emacs/>
- <https://github.com/jceb/vim-orgmode>
- <https://github.com/cit/gpn12-fahrplan.org>

Cheatsheet

- <http://orgmode.org/orgcard.pdf>

Mailingliste

- emacs-orgmode@gnu.org



Vielen Dank!

Fragen?

florian-gpn12@adamsky.it
<http://florian.adamsky.it>
Twitter: @c1t

