

**Get  
productive  
with vimtex  
for LaTeX**

2016/06/16

# Benefits I

---

- PDF viewer of your choice  
[MuPDF](#), [Zathura](#),  
[Okular](#), [qpdfview](#) or  
[SumatraPDF](#)
- **continuous compilation**

# Benefits II

---

- text-objects:
  - ae ... LaTeX environments (e.g. `\begin{itemize}`)
  - ac ... commands
  - i\$ ... inline math structure
  - a\$ ... whole math structure

# Benefits III

---

- motions and mappings:
  - `[[|]]` ... move to next/previous section
  - `%` ... move between matching delimiters
  - `dse|cse` ... delete/change the surrounding environment
  - `dsc|csc` ... delete/change the surrounding command

# Benefits IV

---

- omni completion, improved syntax highlighting and indentation

# Prerequisite I

---

- [latexmk](#) is a perl script that runs the desired/necessary LaTeX command the correct number of times to resolve cross references.
- grab the latest one 22 April 2016. Version 4.45

# Prerequisite II

---

- ensure that `libsynctex` exists in the system for forward and backward searching!

# Prerequisite III

---

- install girara
- make sure zathura is compiled with synctex support



# Install latexmk

---

```
$ cd /tmp
$ wget http://users.phys.psu.edu/%7Ecollins/software/latexmk-jcc/latexmk-445.zip
$ unzip latexmk*.zip
$ sudo cp latexmk/latexmk.pl /usr/local/bin
$ sudo mv /usr/local/bin/latexmk.pl /usr/local/bin/latexmk
```

# Install libsynctex

---

```
#!/bin/bash
cd /tmp && rm -rf libsynctex*

if [ "$(uname -m)" == "x86_64" ]
then
  wget http://de.archive.ubuntu.com/ubuntu/pool/main/t/texlive-bin/libsynctex1_2015.20160222.37495-1_amd64.deb
  wget http://de.archive.ubuntu.com/ubuntu/pool/main/t/texlive-bin/libsynctex-dev_2015.20160222.37495-1_amd64.deb
else
  wget http://de.archive.ubuntu.com/ubuntu/pool/main/t/texlive-bin/libsynctex1_2015.20160222.37495-1_i386.deb
  wget http://de.archive.ubuntu.com/ubuntu/pool/main/t/texlive-bin/libsynctex-dev_2015.20160222.37495-1_i386.deb
fi

sudo dpkg -i libsynctex1* && sudo dpkg -i libsynctex-dev*
```

# Install girara and zathura

---

```
#!/bin/bash
GIRARA_VERSION=0.2.6
ZATHURA_VERSION=0.3.6

rm -rf /tmp/girara /tmp/zathura

# otherwise the own girara compilation will not work
sudo apt-get remove libgirara-dev

# need for zathura compilation
sudo apt-get install libmagic-dev

cd /tmp && git clone https://git.pwmt.org/pwmt/girara.git
cd girara && git checkout $GIRARA_VERSION && make && sudo make install
cd /tmp && git clone https://git.pwmt.org/pwmt/zathura.git
cd zathura && git checkout $ZATHURA_VERSION && make WITH_SYNCTEX=1 && sudo make install
```

# vimtex: Basics

---

- `\lll|:VimtexCompileToggle:`  
compile and it will open the pdf in your preferred pdf-viewer
- watch for `latexmk compile: started` continuous mode in the statusline

# vimtex: set default pdf viewer

---

```
let g:vimtex_view_method = 'zathura'
```

# vimtex: forward and backward search

```
237 \subsection{Rollen}
238 \begin{itemize}
239 \item keine Projektleiter-Rolle in Scrum
240 \item Verantwortlichkeiten des traditionellen Projektleiters sind auf die drei Rollen (Product
241 Owner (PO), ScrumMaster (SM), das Entwicklungs-Team) im Scrum Team aufgeteilt
242 \end{itemize}
243
244
245 \subsection{Das Scrum Team}
246 \begin{itemize}
247 \item besteht aus PO, Entwicklungs-Team und Scrum Master (SM)
248 \item Cross-funktionale Gruppe9(hat alle Fachkräfte, um ein fertiges Produktinkrement am
249 Ende eines Sprints lauffähig zu haben ohne dabei von Personen außerhalb des
250 Entwicklungsteam abhängig zu sein) und selbstorg.
251 \item versucht ein potential auslieferbares Produkt-Inkrement jeden Sprint zu
252 liefern10(ultimative Aufgabe, die Anforderungen des PÖs im Sprint Backlog in ein potenti
253 auslieferbares Produktinkrement zu wandeln) und versuchen, Feedback fürs geschafftes zu maximie
254 \item Teams repräsentieren multilearning11(d.h. jeder hat seine spezielle Stärken, aber es
255 auch Aufgaben in Bereichen erledigt, in denen sie nicht so weit bewandert ist)
256 \item verhandelt Commitments zum Sprint mit PO und legen so den scope des Sprint Backlogs fest
257 \item Schätzen der Größe von Backlog-Einträgen
258 \item hat Autonomie wie die Commitments erreicht werden
259 \item 7 +- zwei Mitglieder12(Team sollte so groß sein, dass es von einer großen Pizza satt
260 wird)
261
```

11. Die besten Architekturen, Anforderungen und Designs entstehen in sich selbst organisierenden Teams.
12. In regelmäßigen Intervallen reflektiert das Team darüber, wie es effektiver werden kann, und passt dann dein Verhalten entsprechend an.

## 1.4 Rollen

- keine Projektleiter-Rolle in Scrum
- Verantwortlichkeiten des traditionellen Projektleiters sind auf die drei Rollen (Product Owner (PO), ScrumMaster (SM), das Entwicklungs-Team) im Scrum Team aufgeteilt

### 1.4.1 Das Scrum Team

- besteht aus PO, Entwicklungs-Team und Scrum Master (SM)
- Cross-funktionale Gruppe<sup>9</sup> und selbstorg.
- versucht ein potential auslieferbares Produkt-Inkrement jeden Sprint zu liefern<sup>10</sup> und versuchen, Feedback fürs geschafftes zu maximieren
- Teams repräsentieren `multilearning`<sup>11</sup>
- verhandelt Commitments zum Sprint mit PO und legen so den scope des Sprint Backlogs fest
- Schätzen der Größe von Backlog-Einträgen
- hat Autonomie wie die Commitments erreicht werden

<sup>9</sup>hat die Fachkräfte, um ein fertiges Produktinkrement am Ende eines Sprints lauffähig zu haben ohne dabei von Personen außerhalb des Entwicklungsteams abhängig zu sein.  
<sup>10</sup>ultimative Aufgabe, die Anforderungen des PÖs im Sprint Backlog in ein potenti

# vimtex: forward search

---

- `\lv|:VimtexView:` Open the generated PDF
- works with [vim'sservername feature](#) or with gvim/MacVim out of the box for zathura

# vimtex: backward search

---

- press <C-Click> on the PDF and you jump right to the place in the terminal.
- works with gvim/MacVim out of the box for zathura



# vimtex: Commands I

---

- `:VimtexTocOpen` | `:VimtexTocToggle`: open a clickable toc in the left pane (q will close the window)
- `:VimtexLabelsOpen` | `:VimtexLabelsToggle`: open table of labels.

# vimtex: Commands II

---

- `:VimtexInfo`: print basic information
- `:VimtexCountWords` | `:VimtexCountLetters`: count the number of words/letters in the document. It will also show the number of math environments, and similar.

# vimtex: Commands III

---

- `:VimtexCompileOutput`: show the output form the compilation command (i.e. from `latexmk`)
- `:VimtexClean`: clean auxilliary files like `*.aux`, `*.out`, etc.  
Use `:VimtexClean!` to remove everything.

# vimtex: Commands IV

---

- `:VimtexErrors:open`  
quickfix window if there are errors or warnings.

# Who am I

---

- [@wikimatze](#)
- writing [@padrinobook](#)
- running [@vimberlin](#)
- updating [@padrinorb](#)
- and organizing [@vim\\_fest](#)