



# ***Dateiverwaltung & Versionskontrolle mit Git***

## ***Einführung***



Inh. Dipl. Ing. Mario Blunk

Buchfinkenweg 3  
99097 Erfurt / Deutschland

Telefon 0176 2904 5844

Email [info@blunk-electronic.de](mailto:info@blunk-electronic.de)

Internet [www.blunk-electronic.de](http://www.blunk-electronic.de)

# Ihre Dateiverwaltung

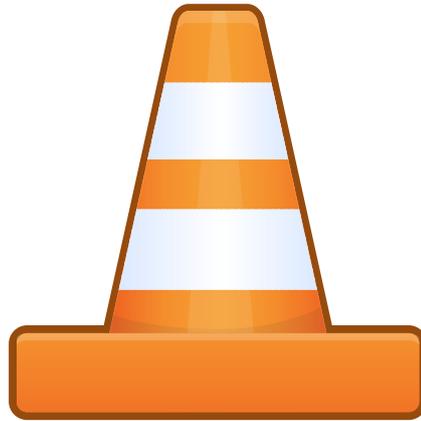


<b>Gegenwärtiger Zustand</b>	<b>Sollzustand</b>
Arbeitsstände in <b>VIELEN</b> Verzeichnissen gespeichert	nur <b>EIN</b> Projektverzeichnis
Arbeitsstand/Version zum Zeitpunkt T ist <b>NICHT</b> einsehbar.	Arbeitsstand/Version zum Zeitpunkt T und dessen Autor <b>JEDERZEIT</b> einsehbar.
Änderungen lassen sich <b>NICHT</b> oder nur sehr <b>UMSTÄNDLICH</b> nachvollziehen.	Änderungen sind in <b>KLARTEXT</b> in Projekthistorie nachvollziehbar.
Änderungen lassen sich <b>NUR</b> mit <b>PROPRIETÄREN</b> Werkzeugen <b>VERSCHIEDENER</b> SW-Applikationen verfolgen.	Änderungen sind mit nur <b>EINEM</b> Werkzeug in <b>ALLEN</b> SW-Applikationen einsehbar.
<b>BINDUNG</b> an SW-Hersteller	<b>KEINE</b> Bindung an SW-Hersteller
Datenhaltung zentral auf <b>EINEM</b> Server	Daten <b>REDUNDANT</b> verteilt auf Rechner der Mitarbeiter.
Mitarbeiter können <b>NUR</b> on-line arbeiten.	Mitarbeiter könne on-line <b>UND</b> off-line arbeiten.
Lizenz, Gebühren, Bindung an Hersteller des Werkzeuges für Dateiverwaltung	<b>KEINE</b> Lizenzen oder Gebühren, <b>KEINE</b> Bindung weil Quellcode offen



# *Überblick*

1. Bestandsaufnahme
2. Was ist Versionskontrolle ?
3. Warum brauchen wir Versionskontrolle ?
4. Was ist Git und was bringt Git ?
5. Beispiele
6. Einschränkungen
7. Fazit

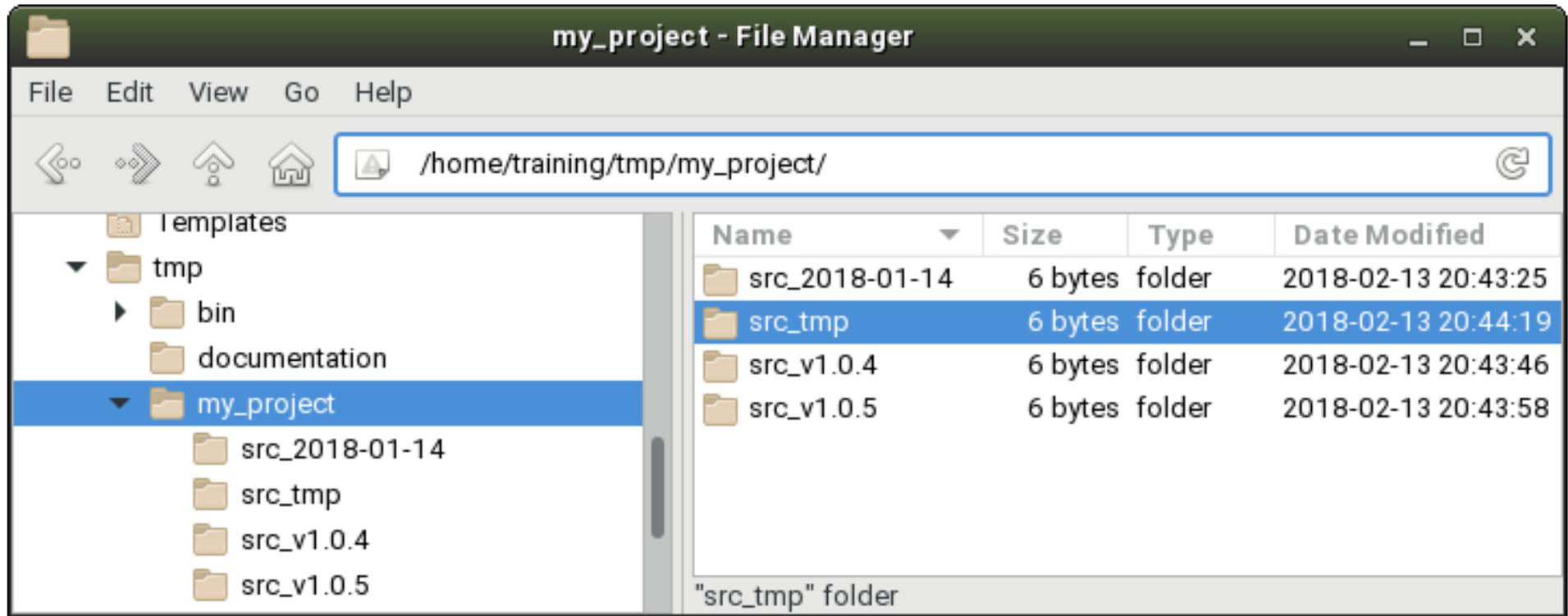


Hinweise für die Anwender am "low end", also Hobbyisten: Einzelkämpfer, kleine bis mittlere Komplexität eines Projekts: Ab wann lohnt es sich, GIT einzusetzen.

# Bestandsaufnahme #1



Wie verwalte ich zur Zeit Dateien ?



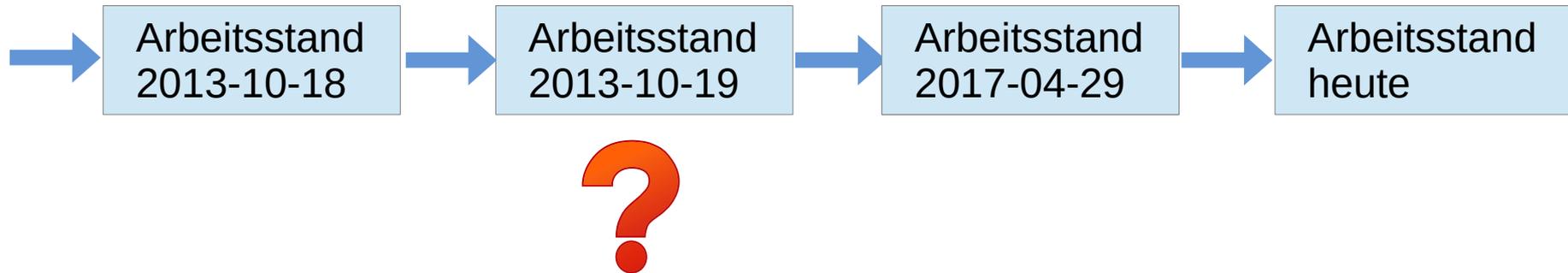
Nachteile:

- Verfolgen von Änderungen **äußerst umständlich**
- Archivierung **bedingt** zuverlässig

# ***Bestandsaufnahme #2***



Kann ich den Arbeitsstand zum Zeitpunkt T einsehen ?

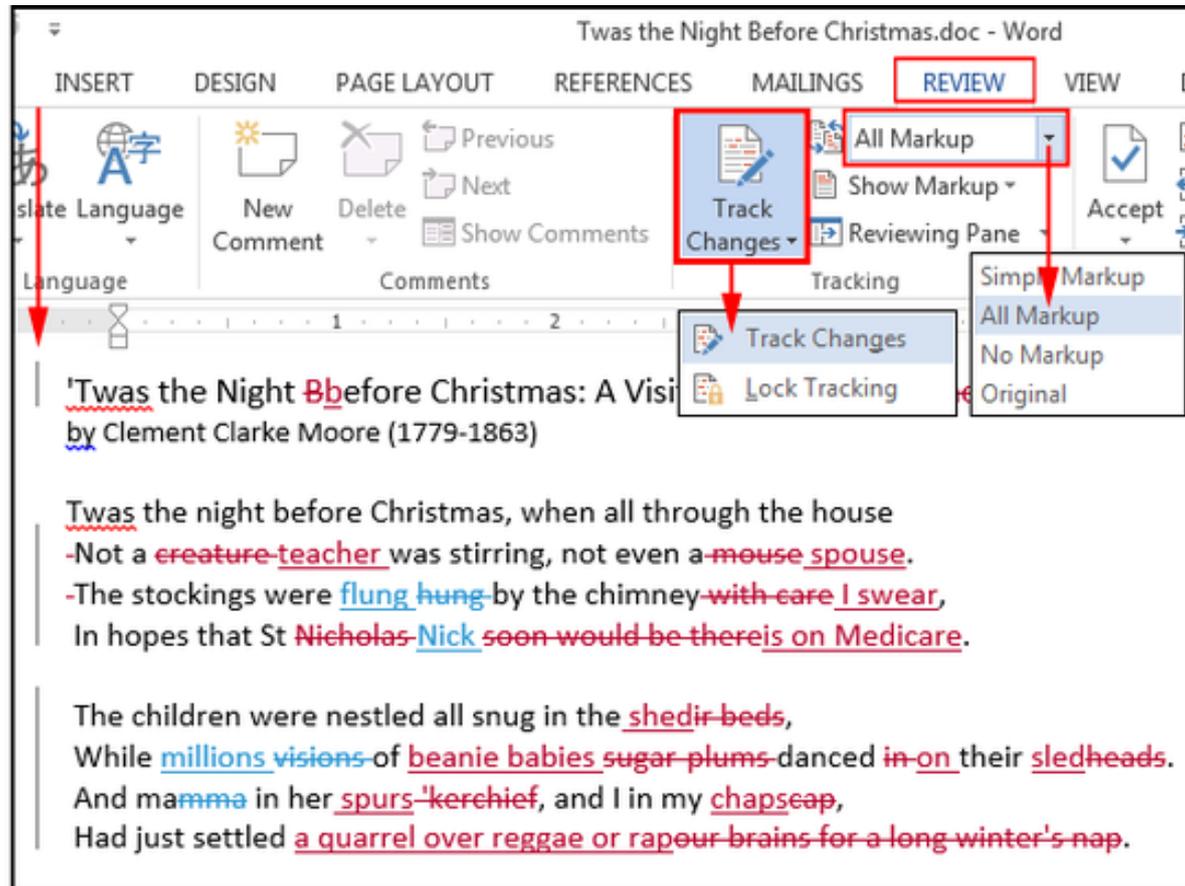


- Wer hat zum Zeitpunkt T was gemacht ?

# Bestandsaufnahme #3a



Kann ich Änderungen unabhängig von der Applikation verfolgen ?



# Bestandsaufnahme #3b



Kann ich Änderungen unabhängig von der Applikation verfolgen ?

## Projekt "Fahrrad"



### Dokumentation

- MS Office
- LibreOffice
- ...



### Mechanik-Konstruktion

- Solid Works
- Autocad
- FreeCad
- ...

```
ld> BC,OFFSET
ld> A,B
ld> I,A>;load
im> 2> ;enabl
```

### Software

- GCC
- MPLAB
- STM32 IDEs
- GPS (GNAT)



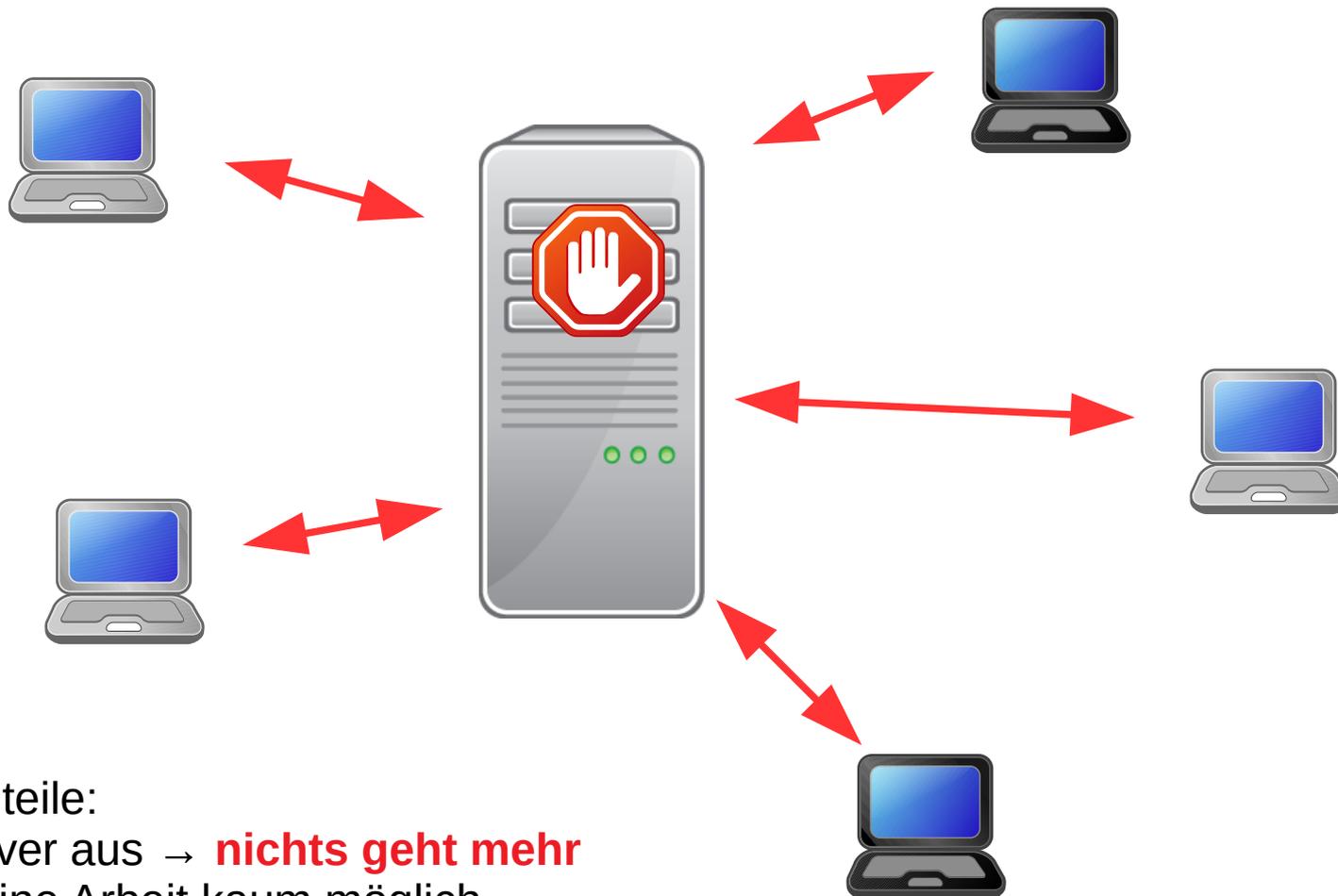
### Elektronik-HW

- KiCad
- EAGLE
- Altium
- Vivado
- ...

# Bestandsaufnahme #4



Sind wir auf einen zentralen Server angewiesen ?



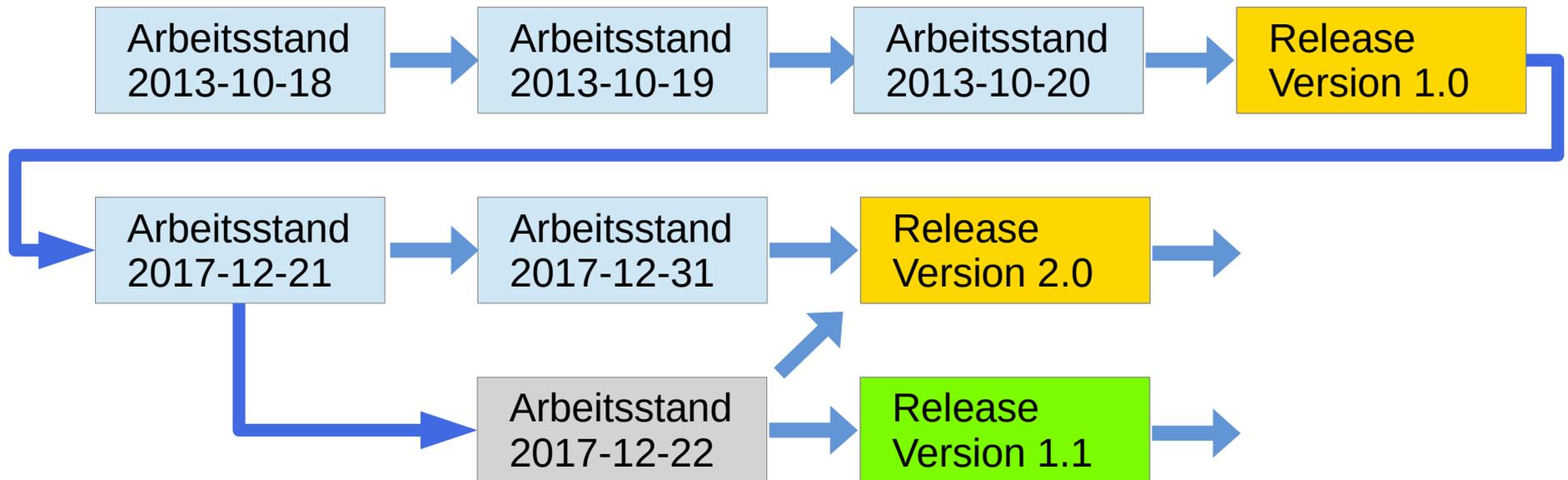
Nachteile:

- Server aus → **nichts geht mehr**
- off-line Arbeit kaum möglich

# Was ist Versionskontrolle ?

Es geht um Quellcode und Klartext (ASCII, XML, ...)

1. Archivieren
2. Verwalten
3. Verfolgen von Änderungen
4. Release/Versionen (wie z.B. V2.16.1, ...)

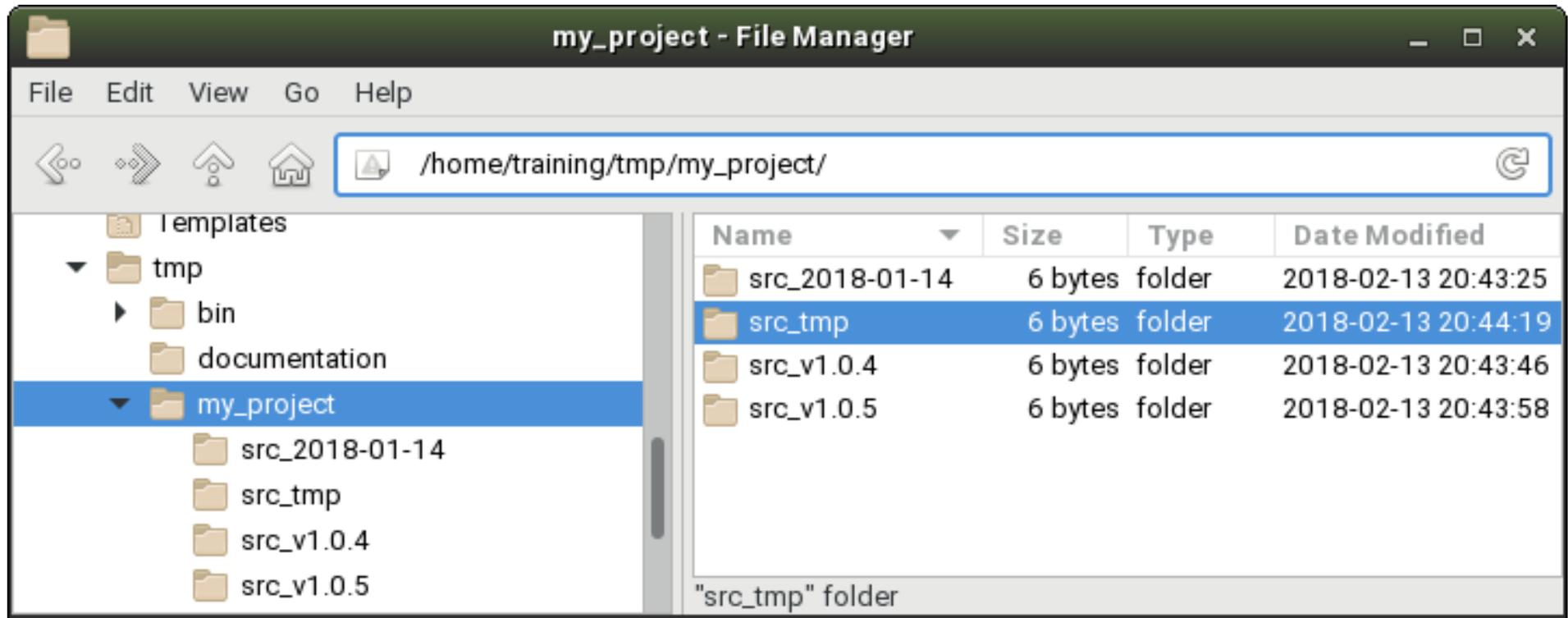


# *Warum Versionskontrolle*



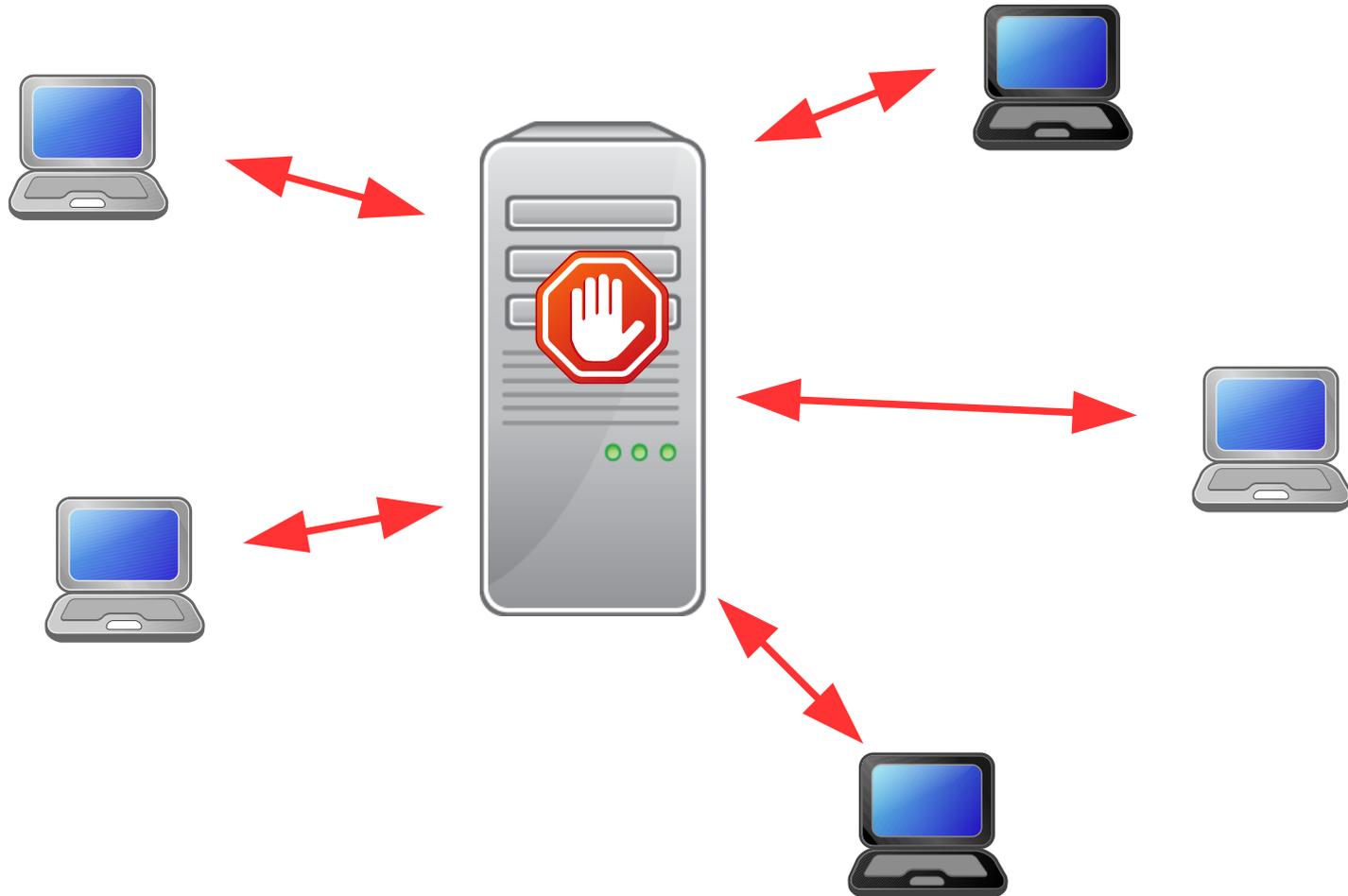
1. erleichterte Fehlersuche/Debugging
2. Archivierung von Konstruktionsdaten (CAD), Quellcode, Dokumentation, Fertigungsdaten (CAM) ...
3. Rückverfolgung von Änderungen
4. für Zertifizierungen (ISO 9001, IEC 61508 / 61511 / 62061, DO-178B, MIL-STD-882-E, ...)

# Verzeichnisbasiert



***So nicht !***

# Zentraler Server

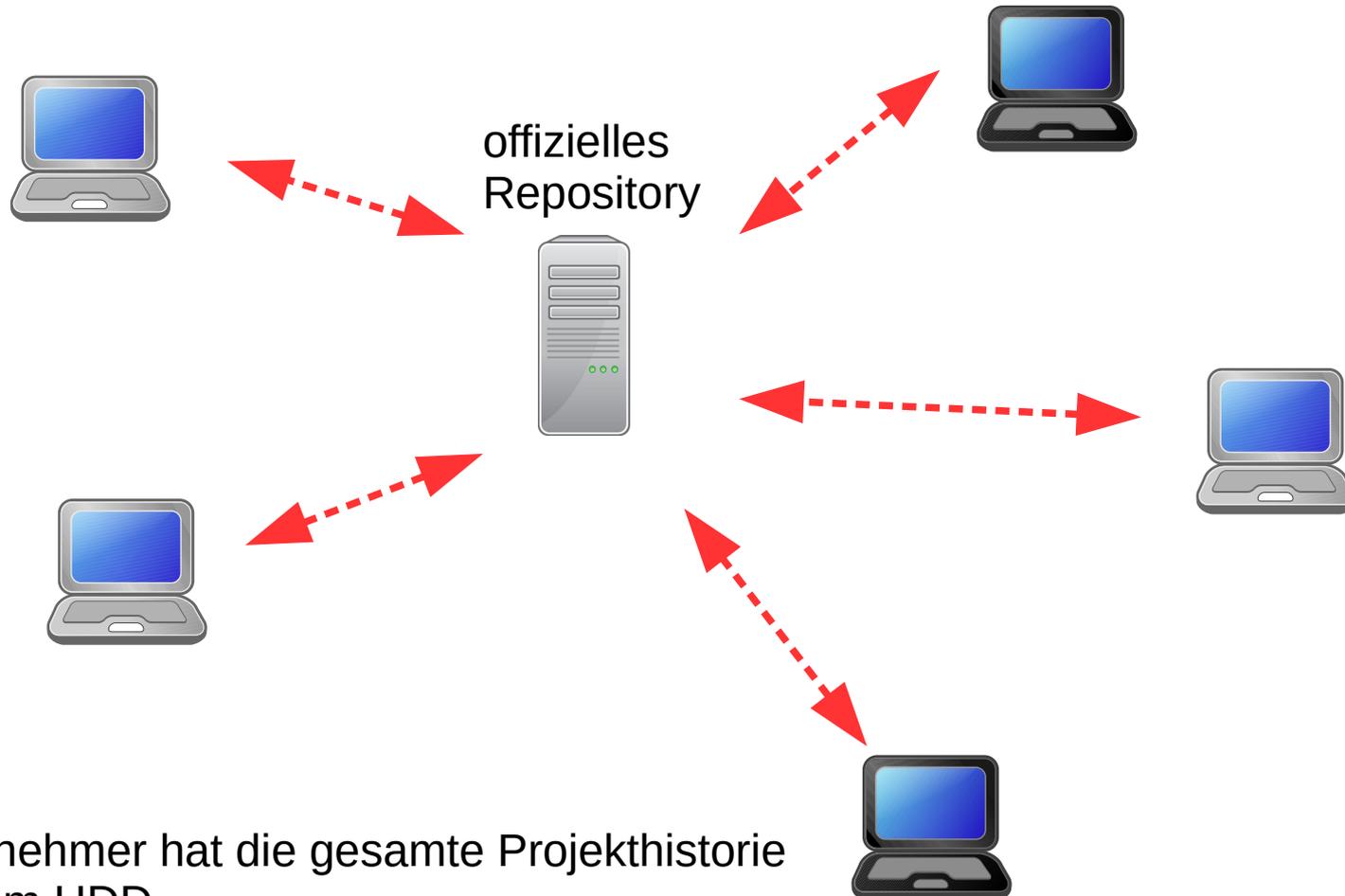


Nachteile:

- Server aus → **nichts geht mehr**
- off-line Arbeit kaum möglich



# *Alles ist lokal - Git*



## Vorteile:

- Jeder Teilnehmer hat die gesamte Projekthistorie auf lokalem HDD
- off-line Arbeit möglich
- Datensicherheit
- OpenSource (OSS)

# Beispiel #1a Assembler

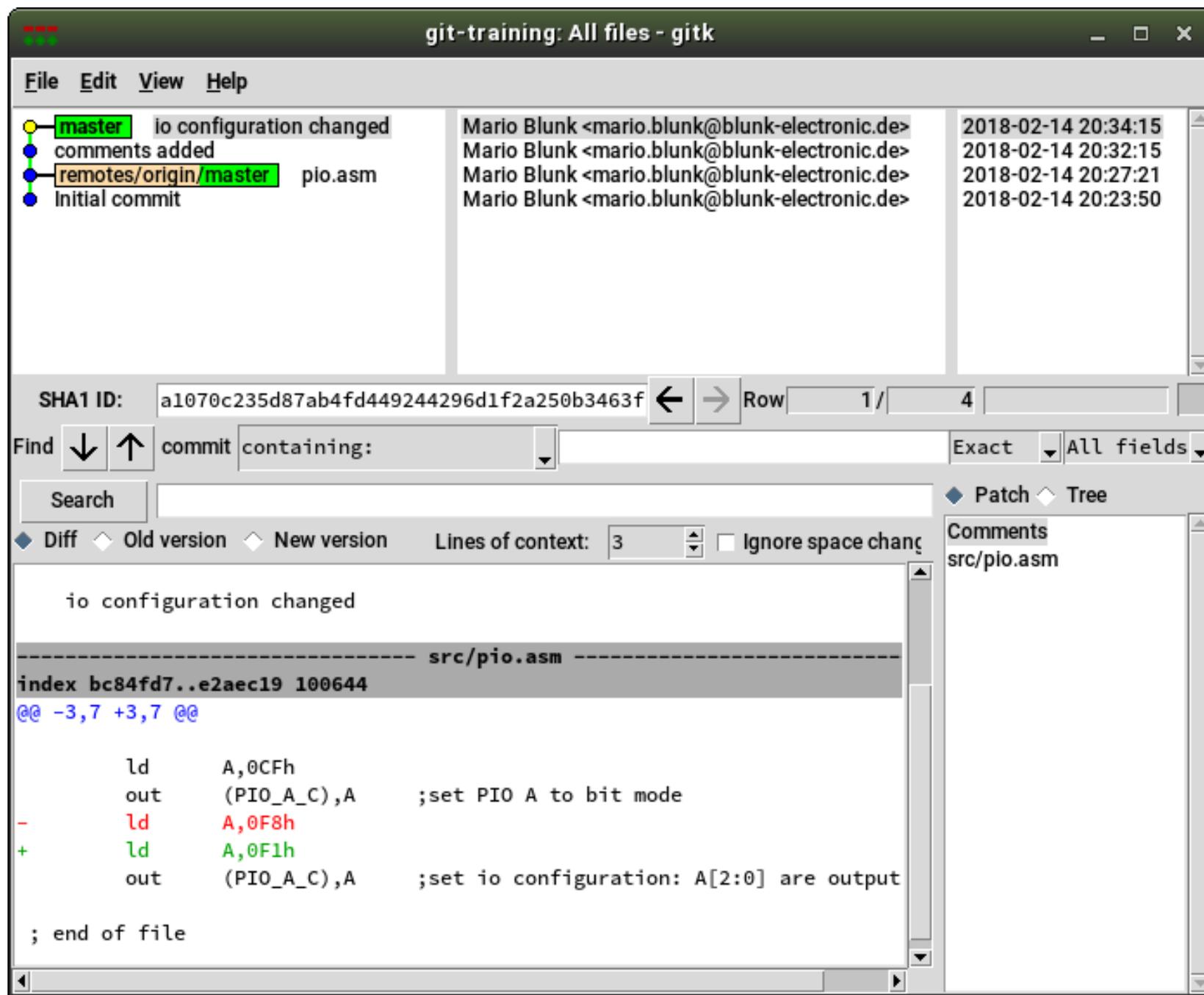
*ursprünglicher Code*

```
ld    A,0CFh
out   (PIO_A_C),A    ;set PIO A to bit mode
ld    A,0F8h
out   (PIO_A_C),A    ;set io configuration: A[2:0] are outputs
```

*geänderter Code*

```
ld    A,0CFh
out   (PIO_A_C),A    ;set PIO A to bit mode
ld    A,0F1h
out   (PIO_A_C),A    ;set io configuration: A[2:0] are outputs
```

# Beispiel #1b Assembler



The screenshot shows a gitk window titled "git-training: All files - gitk". The top panel displays the commit history for the "pio.asm" file. The commit list shows four commits, all by Mario Blunk, with the following messages and timestamps:

Commit Message	Author	Date
io configuration changed	Mario Blunk <mario.blunk@blunk-electronic.de>	2018-02-14 20:34:15
comments added	Mario Blunk <mario.blunk@blunk-electronic.de>	2018-02-14 20:32:15
remotes/origin/master pio.asm	Mario Blunk <mario.blunk@blunk-electronic.de>	2018-02-14 20:27:21
Initial commit	Mario Blunk <mario.blunk@blunk-electronic.de>	2018-02-14 20:23:50

The bottom panel shows a diff view for the file "src/pio.asm". The diff highlights changes between two versions of the file. The current version (index bc84fd7..e2aec19) shows the following changes:

```
index bc84fd7..e2aec19 100644
@@ -3,7 +3,7 @@
     ld    A,0CFh
     out  (PIO_A_C),A    ;set PIO A to bit mode
-    ld    A,0F8h
+    ld    A,0F1h
     out  (PIO_A_C),A    ;set io configuration: A[2:0] are output

; end of file
```

A red arrow points to the diff view.

# Beispiel #1c Assembler

# letzten Stand vom offiziellen Repo holen  
\$ **git pull**

Arbeiten: zeichnen, programmieren, dokumentieren

# gegenwärtigen Dateiinhalt auf Index setzen  
\$ **git add pio.asm**

# Änderung einchecken  
\$ **git commit -m "io configuration geändert"**

# neuen Stand zum offiziellen Repo senden  
\$ **git push**



auch mit GUI verfügbar (Linux, Windows, Mac, ...)

# Beispiel #2a Hochsprache C

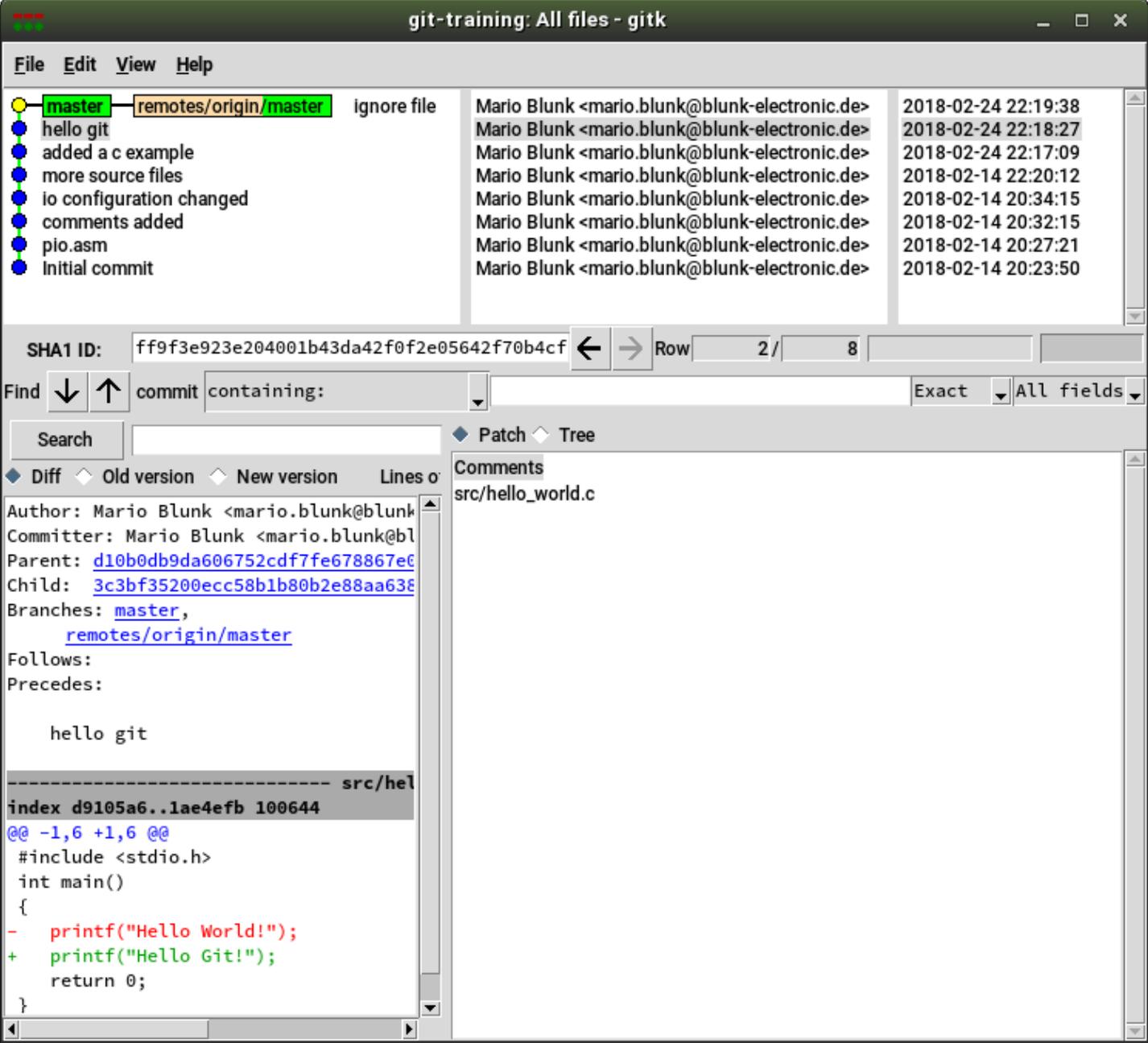
*ursprünglicher Code*

```
#include <stdio.h>
int main()
{
    printf("Hello World!");
    return 0;
}
```

*geänderter Code*

```
#include <stdio.h>
int main()
{
    printf("Hello Git!");
    return 0;
}
```

# Beispiel #2b Hochsprache C



The screenshot shows a Git GUI window titled "git-training: All files - gitk". The window is divided into several sections:

- Commit History:** A list of commits on the left, with the current commit highlighted in green. The list includes "Initial commit", "pio.asm", "comments added", "io configuration changed", "more source files", "added a c example", "hello git", and "ignore file".
- Commit Details:** A table on the right showing commit details for the selected commit, including the author (Mario Blunk), email (mario.blunk@blunk-electronic.de), and timestamp (2018-02-24 22:19:38).
- SHA1 ID:** A text field showing the SHA1 ID of the selected commit: ff9f3e923e204001b43da42f0f2e05642f70b4cf.
- Find:** A search bar with the text "commit containing:" and a dropdown menu set to "Exact".
- Diff View:** A diff view showing the changes between the parent and child commits. The diff is for the file "src/hello\_world.c". The changes are highlighted in red (deletion) and green (addition).

The diff view shows the following changes:

```
index d9105a6..1ae4efb 100644
@@ -1,6 +1,6 @@
#include <stdio.h>
int main()
{
- printf("Hello World!");
+ printf("Hello Git!");
return 0;
}
```



# Beispiel #3 Verilog HDL

*ursprünglicher Code*

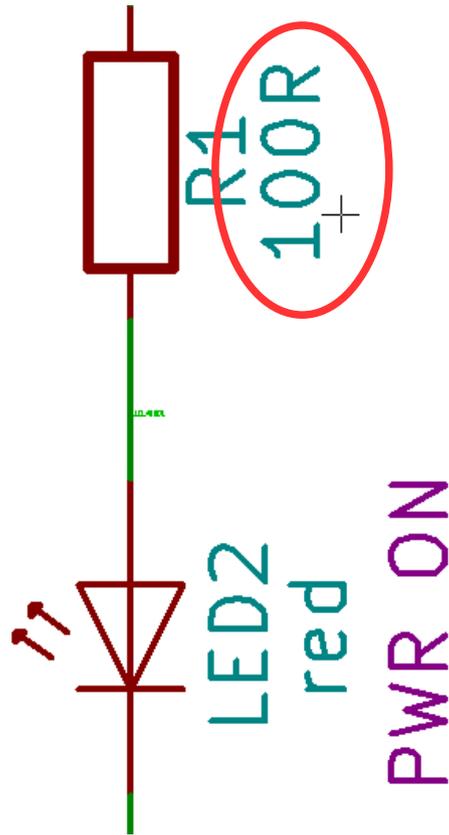
```
module i2c_master(  
    clk,          // input  
    reset,       // input synchronous  
    sda,         // inout  
    scl,         // inout
```

*geänderter Code*

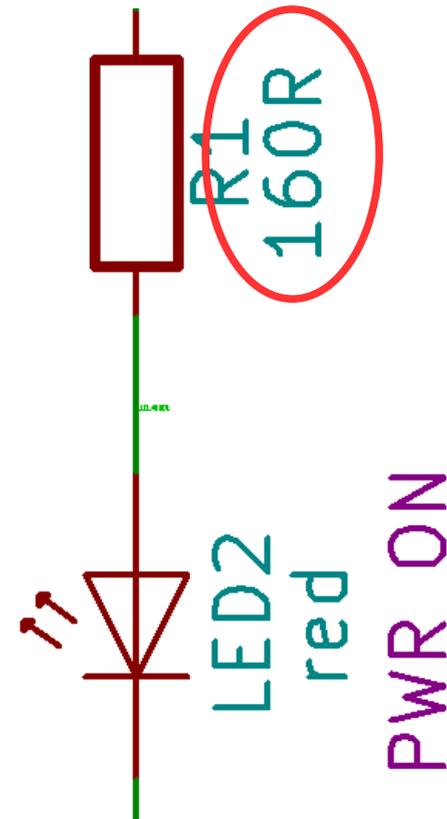
```
module i2c_master(  
    clk,          // input  
    reset_n,     // input asynchronous  
    reset,       // input synchronous  
    sda,         // inout  
    scl,         // inout
```

# Beispiel #4a KiCad

ursprünglicher Schaltplan

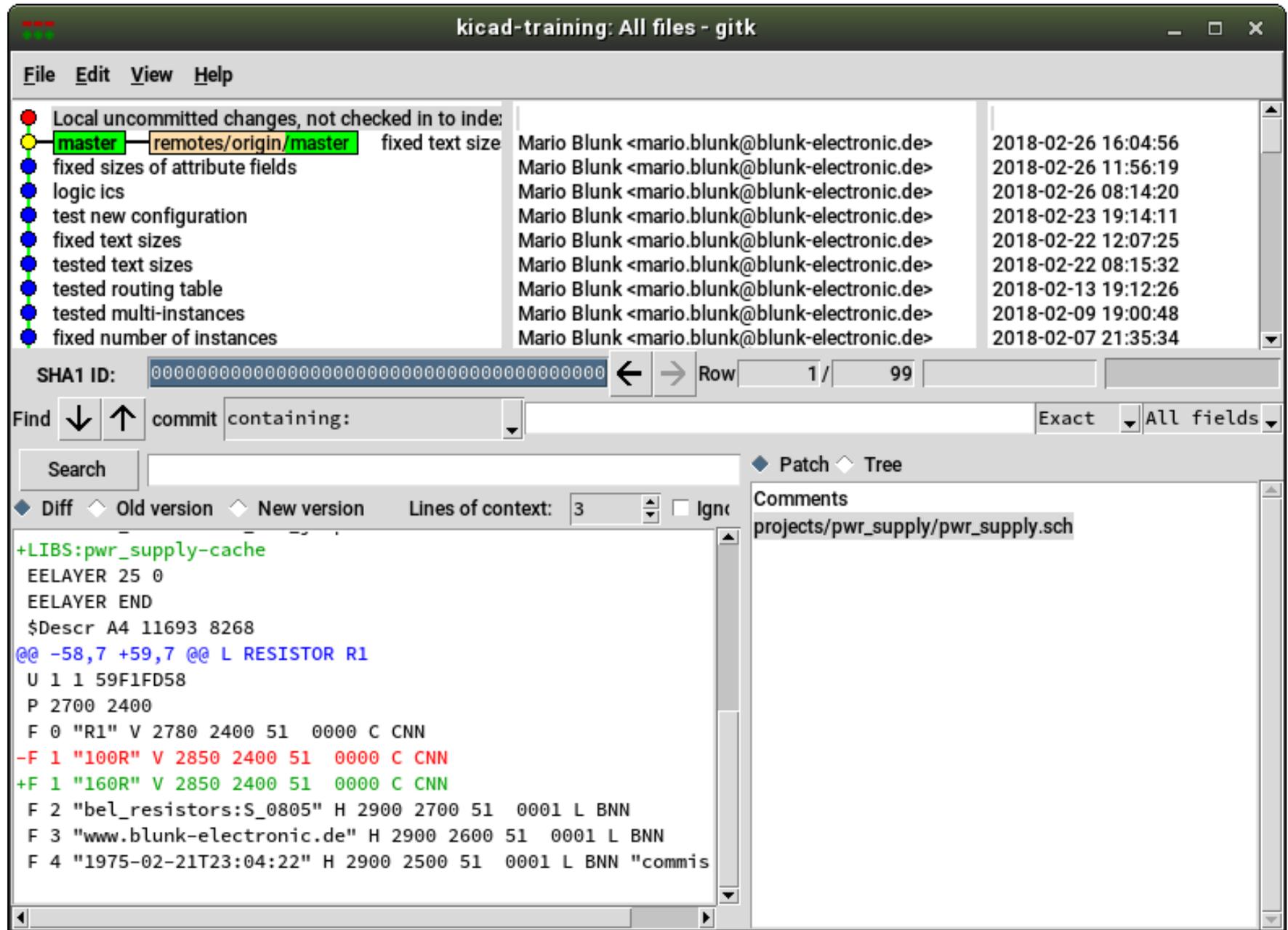


geänderter Schaltplan



auch für Materiallisten (in \*.csv) verwendbar

# Beispiel #4b KiCad



The screenshot shows the gitk GUI for a repository named "kicad-training". The top panel displays the commit history, with the current commit highlighted. The commit message is "fixed text size". The commit log shows the following entries:

SHA1 ID	Author	Date
00	Mario Blunk <mario.blunk@blunk-electronic.de>	2018-02-26 16:04:56
	Mario Blunk <mario.blunk@blunk-electronic.de>	2018-02-26 11:56:19
	Mario Blunk <mario.blunk@blunk-electronic.de>	2018-02-26 08:14:20
	Mario Blunk <mario.blunk@blunk-electronic.de>	2018-02-23 19:14:11
	Mario Blunk <mario.blunk@blunk-electronic.de>	2018-02-22 12:07:25
	Mario Blunk <mario.blunk@blunk-electronic.de>	2018-02-22 08:15:32
	Mario Blunk <mario.blunk@blunk-electronic.de>	2018-02-13 19:12:26
	Mario Blunk <mario.blunk@blunk-electronic.de>	2018-02-09 19:00:48
	Mario Blunk <mario.blunk@blunk-electronic.de>	2018-02-07 21:35:34

The bottom panel shows the diff view for the selected commit. The diff is for the file "projects/pwr\_supply/pwr\_supply.sch". The changes are as follows:

```
+LIBS:pwr_supply-cache
EELAYER 25 0
EELAYER END
$Descr A4 11693 8268
@@ -58,7 +59,7 @@ L RESISTOR R1
U 1 1 59F1FD58
P 2700 2400
F 0 "R1" V 2780 2400 51 0000 C CNN
-F 1 "100R" V 2850 2400 51 0000 C CNN
+F 1 "160R" V 2850 2400 51 0000 C CNN
F 2 "bel_resistors:S_0805" H 2900 2700 51 0001 L BNN
F 3 "www.blunk-electronic.de" H 2900 2600 51 0001 L BNN
F 4 "1975-02-21T23:04:22" H 2900 2500 51 0001 L BNN "commis
```

A red arrow points to the diff view.

# ***Einschränkungen***

**Es geht um Quellcode und Klartext (ASCII, XML, ...)**

Änderungsverfolgung nicht möglich mit:

\*.doc / \*.docx

\*.ppt / \*.pptx

\*.odt / \*.ods (statt dessen \*.fodt / \*.fodp verwenden)

\*.schdoc

\*.pcbdoc

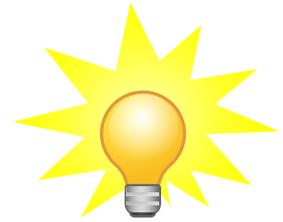
\*.bin

\*.jpg / \*.bmp ...

\*.wav / \*.mp3 / \*.ogg

.....

# ***Fazit***



1. Archivieren
2. Verwalten
3. Verfolgen von Änderungen → vereinfachte Fehlersuche
4. wenn OpenSource Werkzeuge verwendet werden → keine Bindung an SW-Hersteller
5. dezentrale Datenhaltung → Redundanz → Datensicherheit
6. Projektteilnehmer können off-line arbeiten
7. kostenfreies Werkzeug zur Versionskontrolle



# Links

[http://www.blunk-electronic.de/pdf/git\\_training\\_teil\\_1.pdf](http://www.blunk-electronic.de/pdf/git_training_teil_1.pdf)



<https://de.wikipedia.org/wiki/Git>

<https://de.wikipedia.org/wiki/GitHub>

<https://de.wikipedia.org/wiki/GitLab>

<https://de.wikipedia.org/wiki/Bitbucket>

<https://git-scm.com/doc>

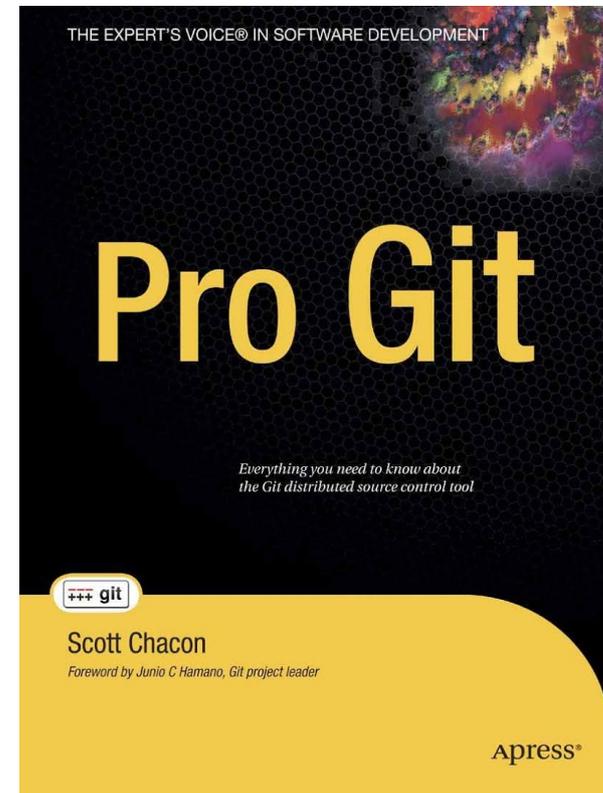
<https://github.com/Blunk-electronic>

# Literatur

<http://gitbu.ch/>



<https://git-scm.com/book/de/v1>  
<https://git-scm.com/book/en/v2>



<https://progit2.s3.amazonaws.com/en/2016-03-22-f3531/progit-en.1084.pdf>

***Danke für Ihre Aufmerksamkeit !***