Dateiverwaltung
&
Versionskontrolle
mit Git

Einführung

Info für Manager: Lesezeit 10min

Doc. Vers. 2018-03-22

Inh. Dipl. Ing. Mario Blunk

Buchfinkenweg 3
99097 Erfurt / Deutschland

Telefon 0176 2904 5844

Email info@blunk-electronic.de

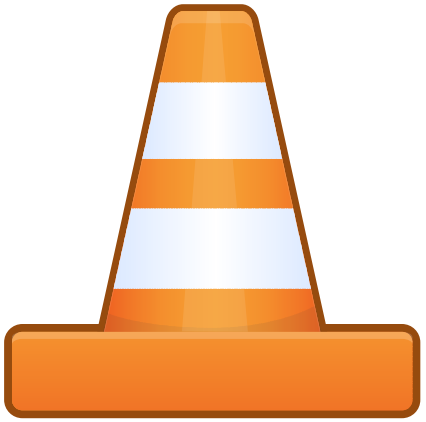
Internet www.blunk-electronic.de

Ihre Dateiverwaltung

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

Überblick

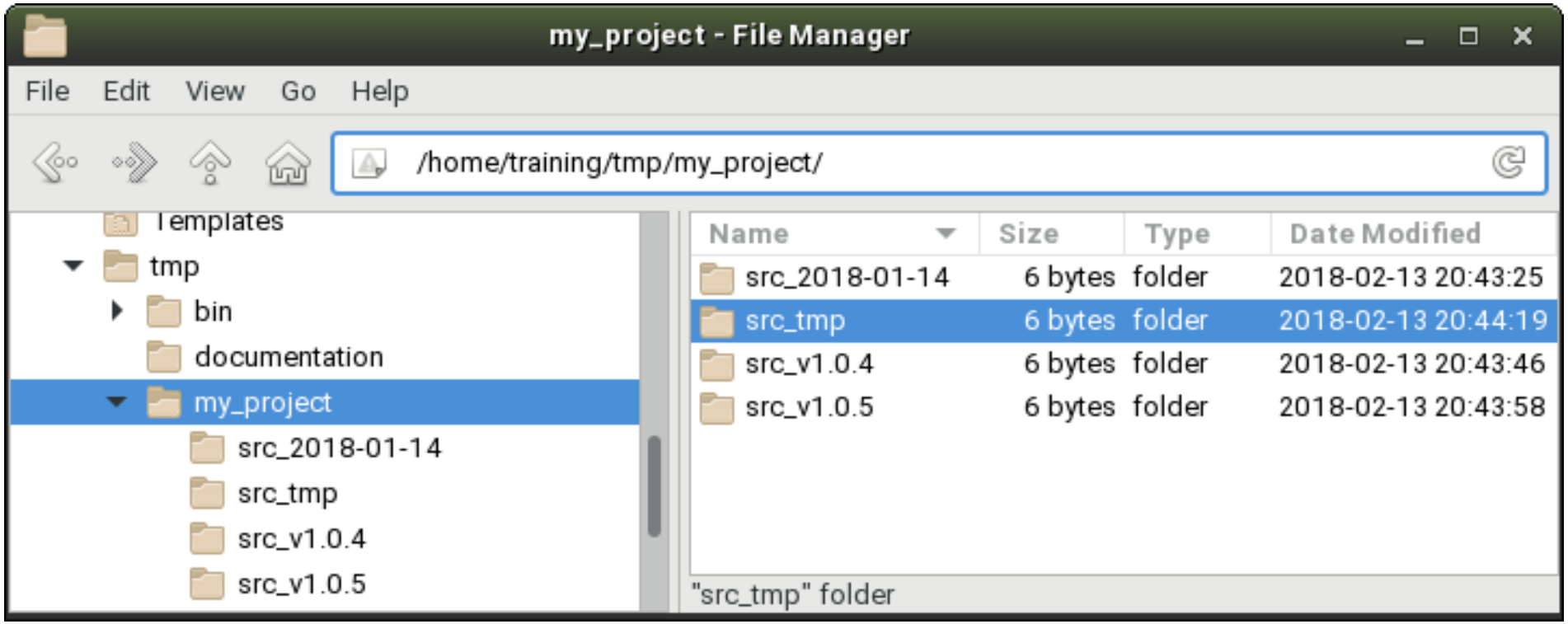
1. Bestandsaufnahme
2. Was ist Versionskontolle ?
3. Warum brauchen wir Versionskontrolle ?
4. Was ist Git und was bringt Git ?
5. Beispiele
6. Einschränkungen
7. Fazit



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

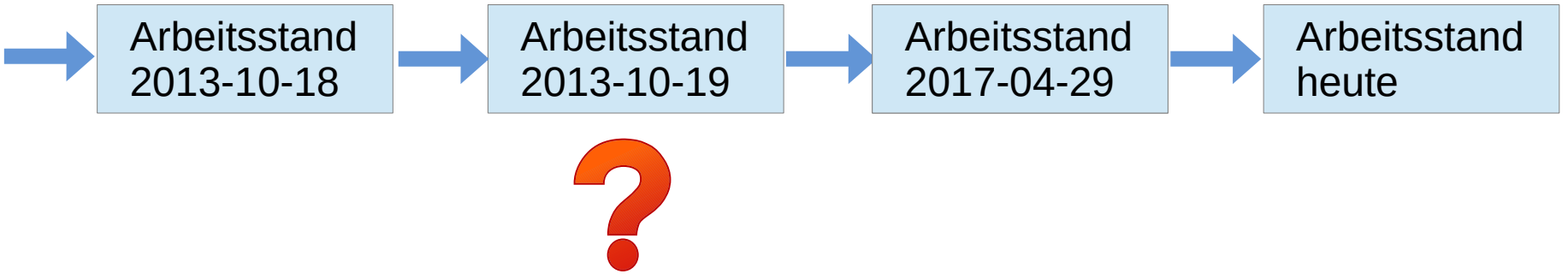
Wie verwalte ich zur Zeit Dateien ?



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

Bestandsaufnahme #2

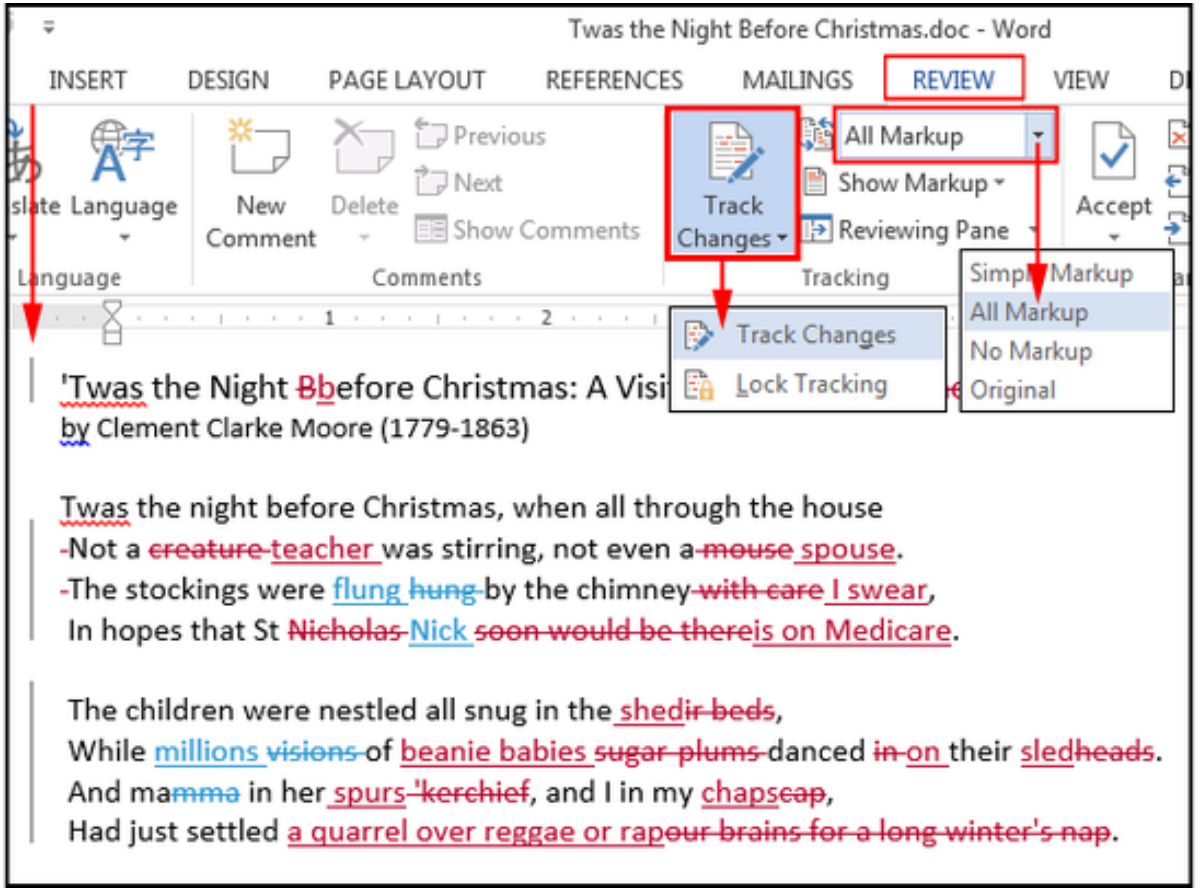
Kann ich den Arbeitsstand zum Zeitpunkt T einsehen ?



- Wer hat zum Zeitpunkt T was gemacht ?

Bestandsaufnahme #3a

Kann ich Änderungen unabhängig von der Applikation verfolgen ?



https://www.pcworld.com/article/2861859/how-to-track-changes-in-microsoft-word-without-going-insane.html

Bestandsaufnahme #3b

Kann ich Änderungen unabhängig von der Applikation verfolgen ?

Projekt “Fahrrad”

Dokumentation
- MS Office
- LibreOffice
- ...

Mechanik-Konstruktion
- Solid Works
- Autocad
- FreeCad
- ...

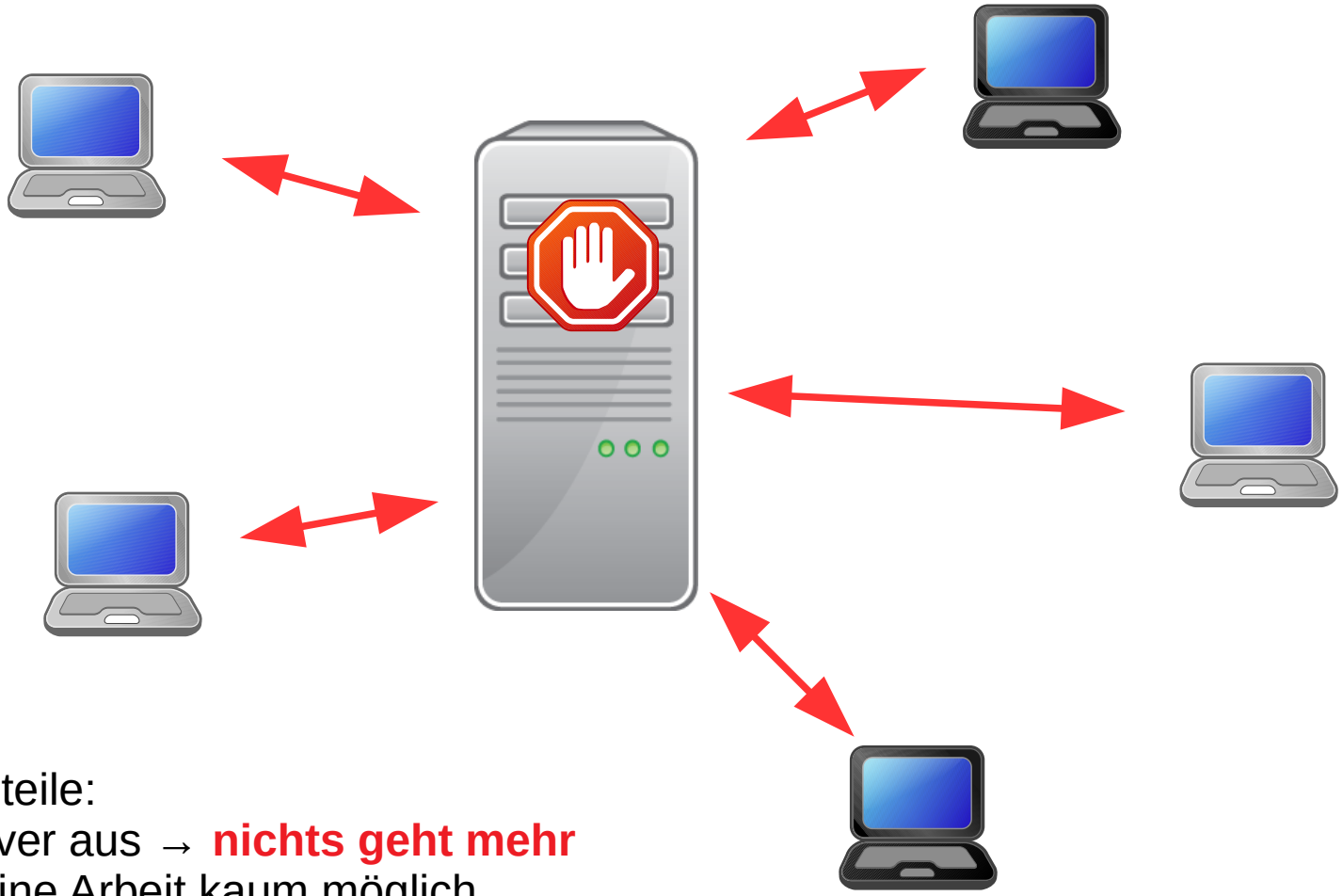
```
ld» BC,OFFSET
ld» A,B
ld» I,A»;load
im» 2»   ;enabl
```

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

Elektronik-HW
- KiCad
- EAGLE
- Altium
- Vivado
- ...

Bestandsaufnahme #4

Sind wir auf einen zentralen Server angewiesen ?

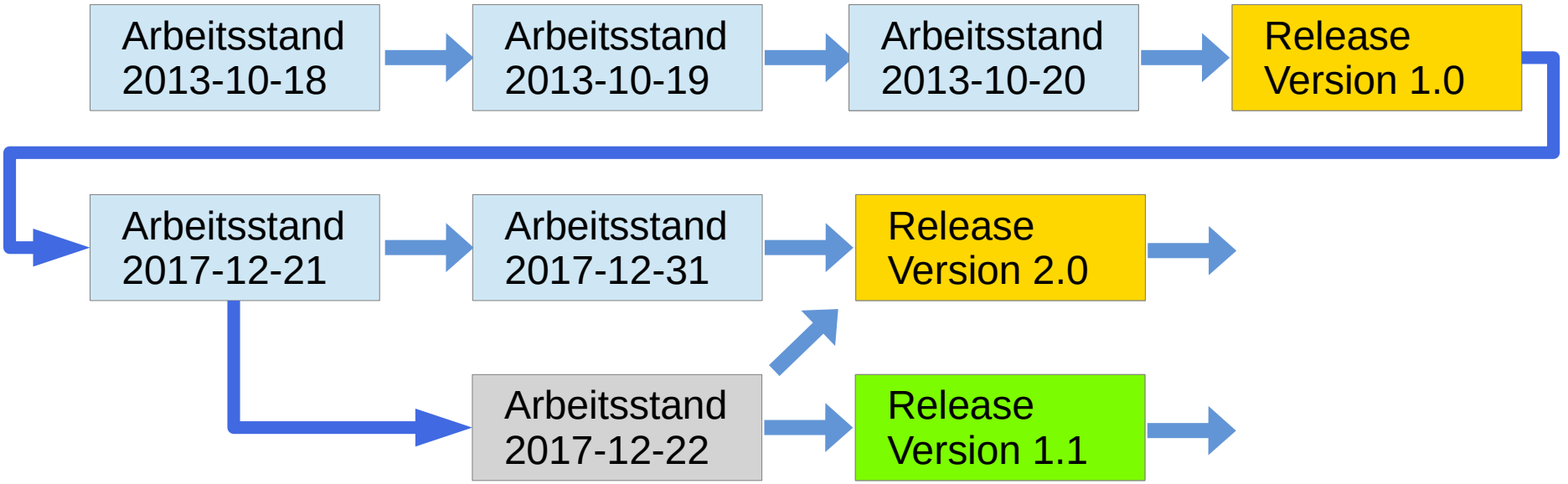


Nachteile:
- Server aus → **nichts geht mehr**
- off-line Arbeit kaum möglich

Was ist Versionskontrolle ?

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

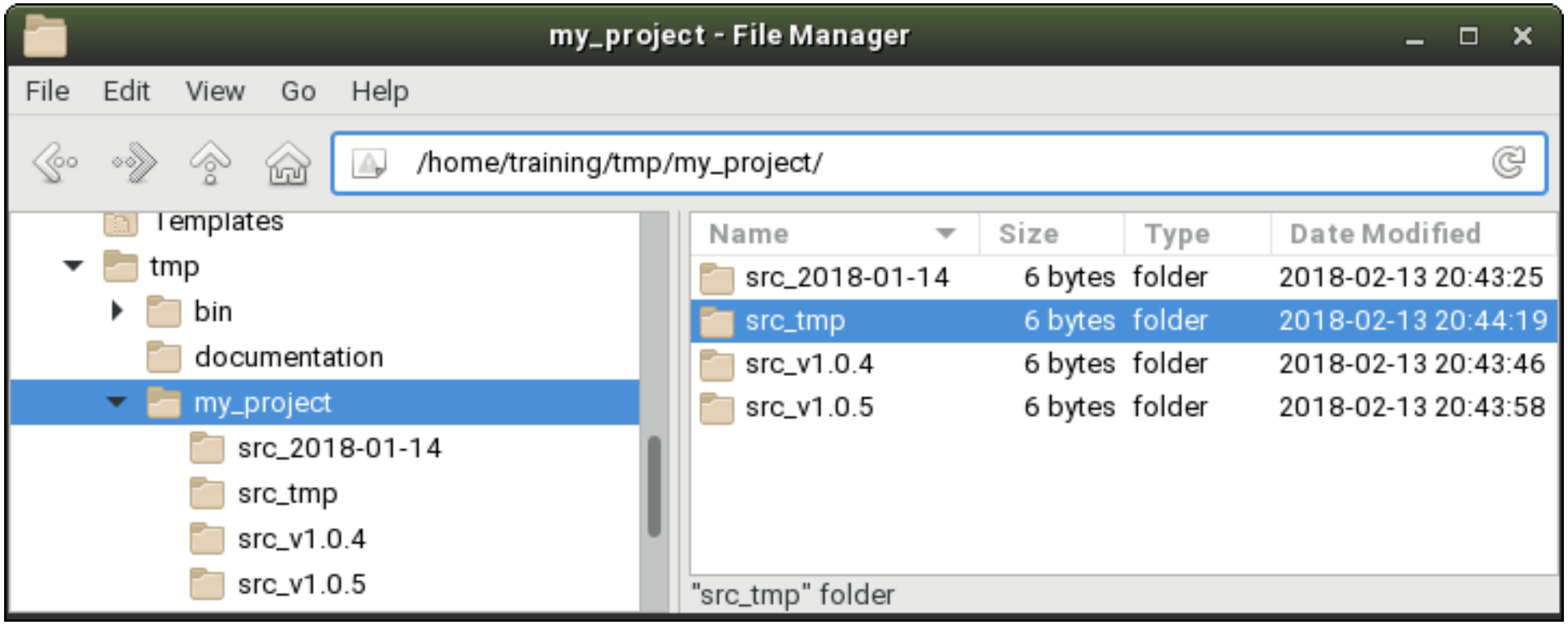
1. Archivieren
2. Verwalten
3. Verfolgen von Änderungen
4. Release/Versionen (wie z.B. V2.16.1, ...)



Warum Versionskontrolle

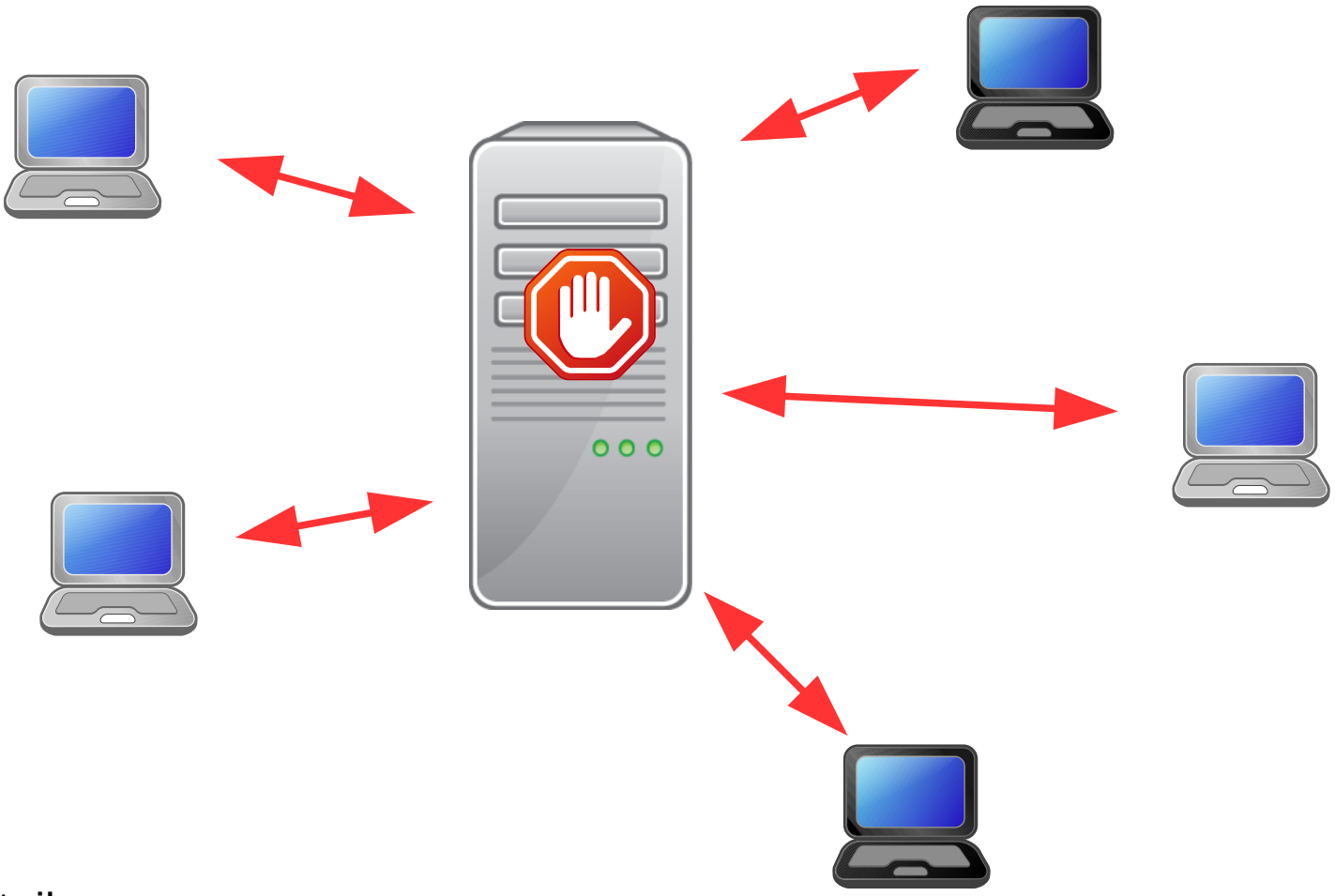
1. erleichterte Fehlersuche/Debugging
2. Archivierung von Konstruktionsdaten (CAD), Quellcode, Dokumentation, Fertigungsdaten (CAM) ...
3. Rückverfolgung von Änderungen
4. für Zertifizierungen (ISO 9001, IEC 61508 / 61511 / 62061, DO-178B, MIL-STD-882-E, ...)

Verzeichnisbasiert



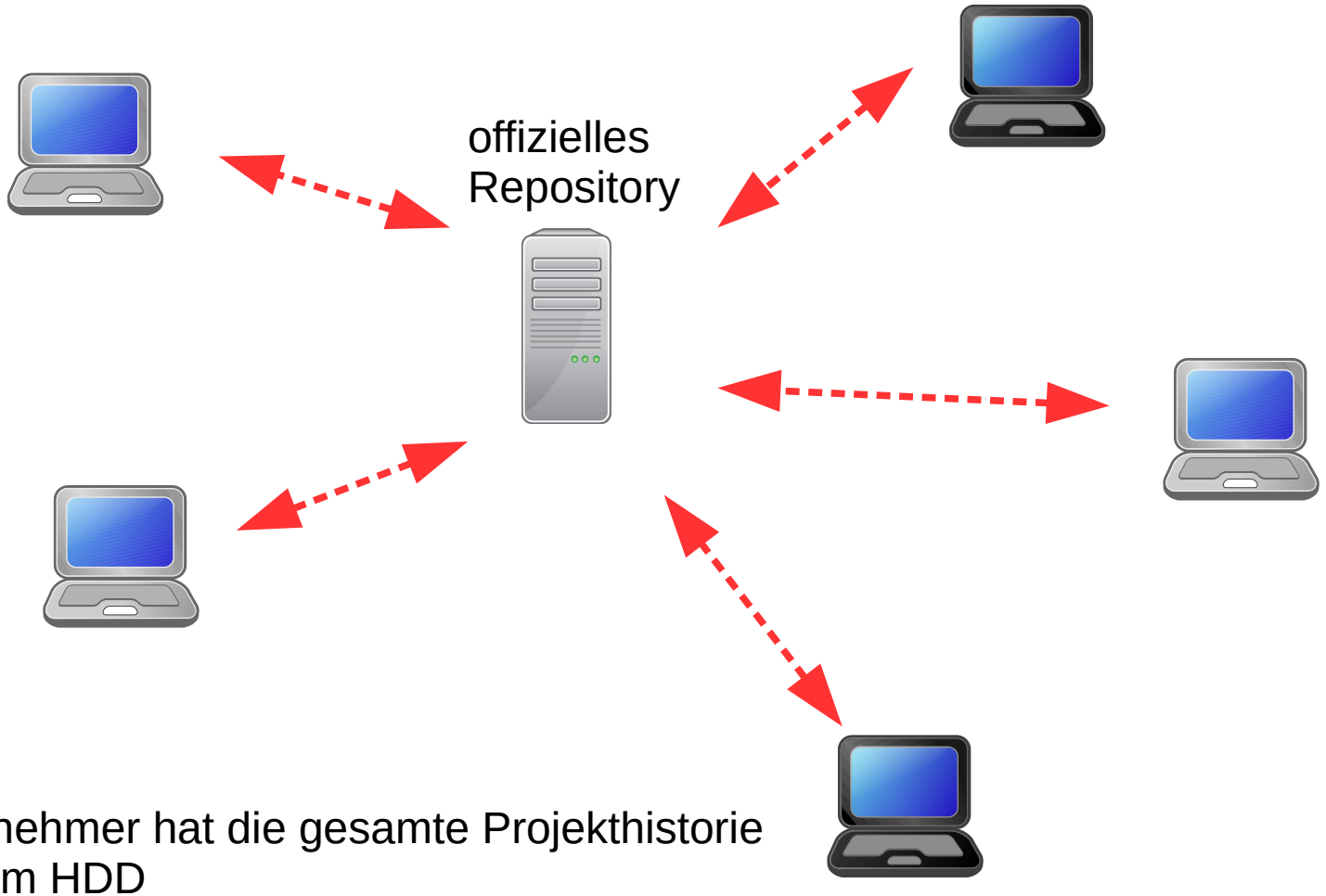
So nicht !

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

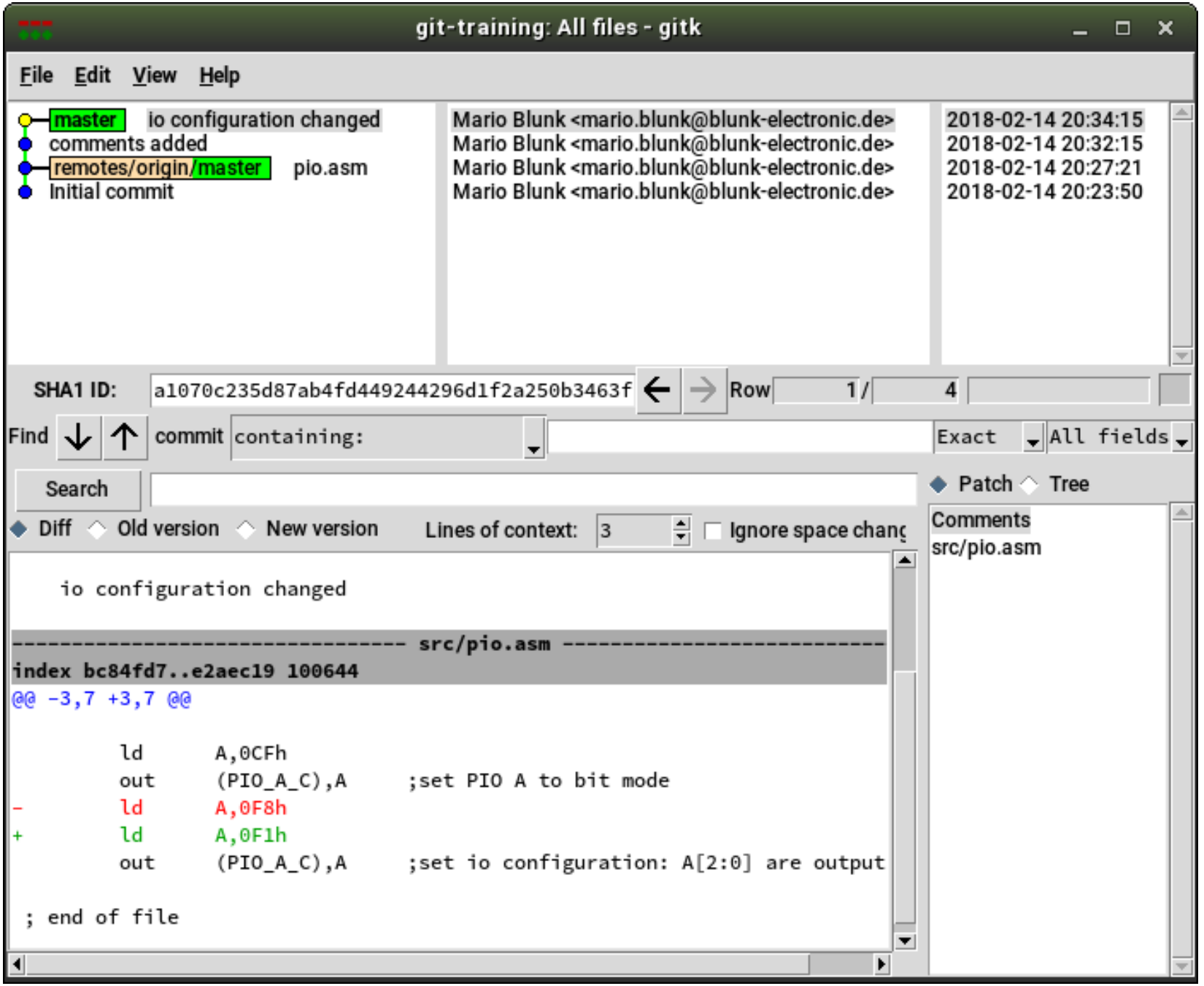
```
ld      A,0CFh
out     (PIO_A_C),A     ;set PIO A to bit mode
ld      A,0F8h
out     (PIO_A_C),A     ;set io configuration: A[2:0] are outputs
```

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
```

https://github.com/Blunk-electronic/git-training

Beispiel #1b Assembler



Beispiel #1c Assembler

```
# letzten Stand vom offizellen 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 offizellen Repo senden
$ git push
```


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

Beispiel #2a Hochsprache C

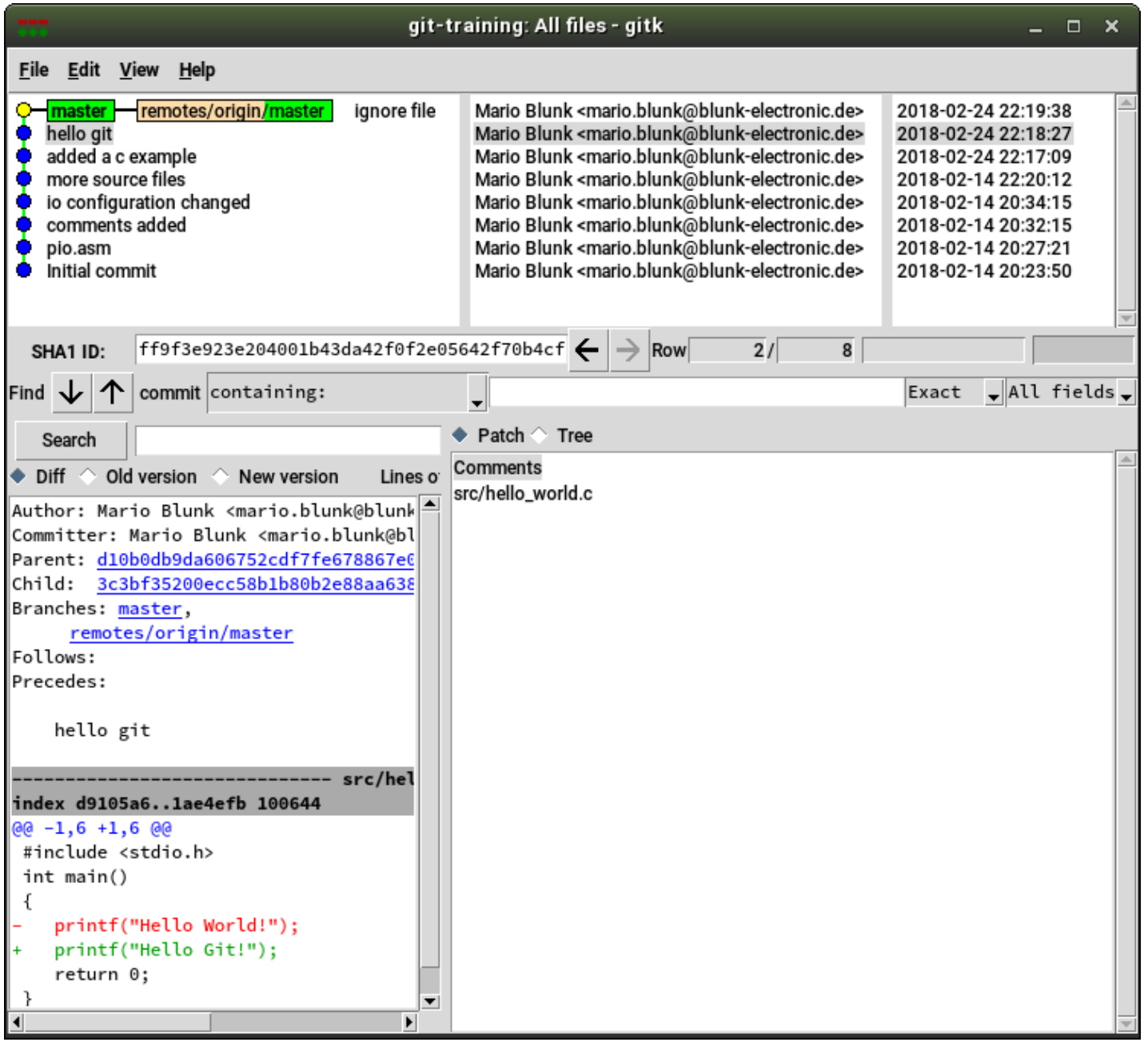
ursprünglicher Code

```
#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



Beispiel #3 Verilog HDL

ursprünglicher Code

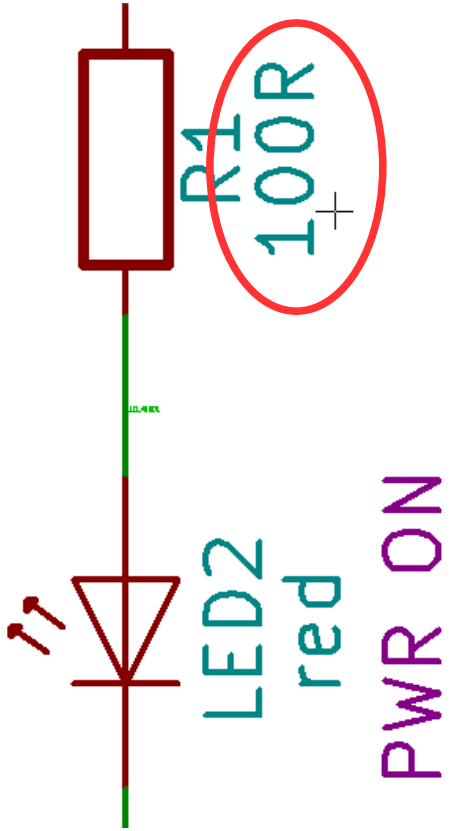
```
module i2c_master(
    clk,          // input
    reset,        // input synchronous
    sda,          // inout
    scl,          // inout
```

geänderter Code

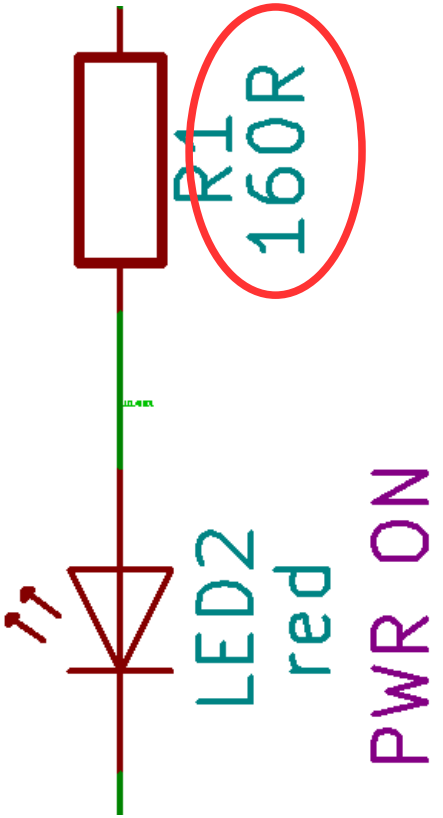
```
module i2c_master(
    clk,          // input
    reset_n,      // input asynchronous
    reset,        // input synchronous
    sda,          // inout
    scl,          // inout
```

Beispiel #4a KiCad

ursprünglicher Schaltplan

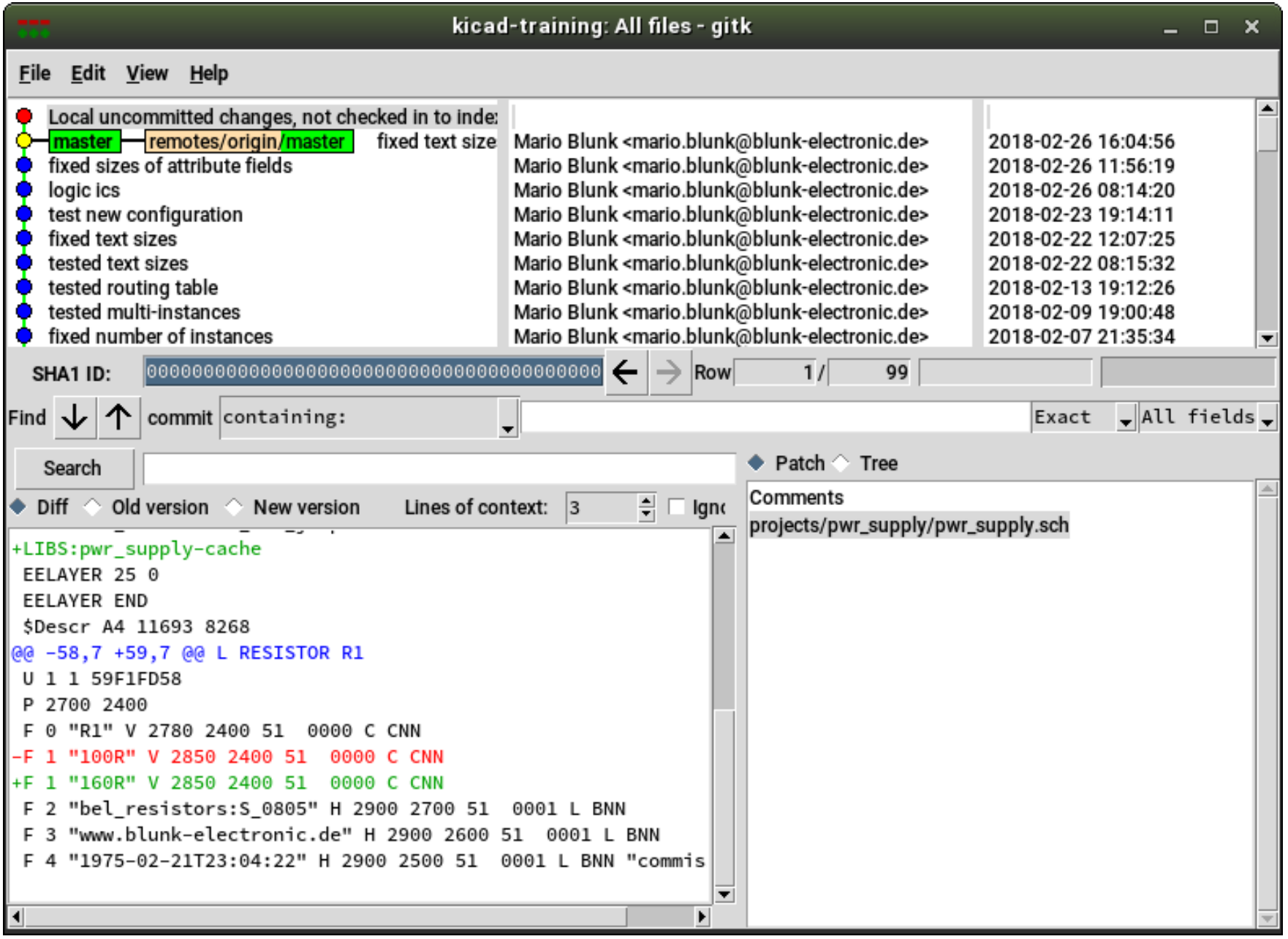


geänderter Schaltplan



auch für Materiallisten (in *.csv) verwendbar

Beispiel #4b KiCad



Einschänkungen

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

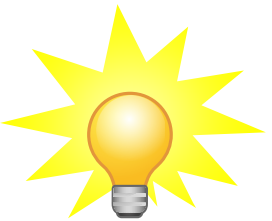
*.bin

*.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

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

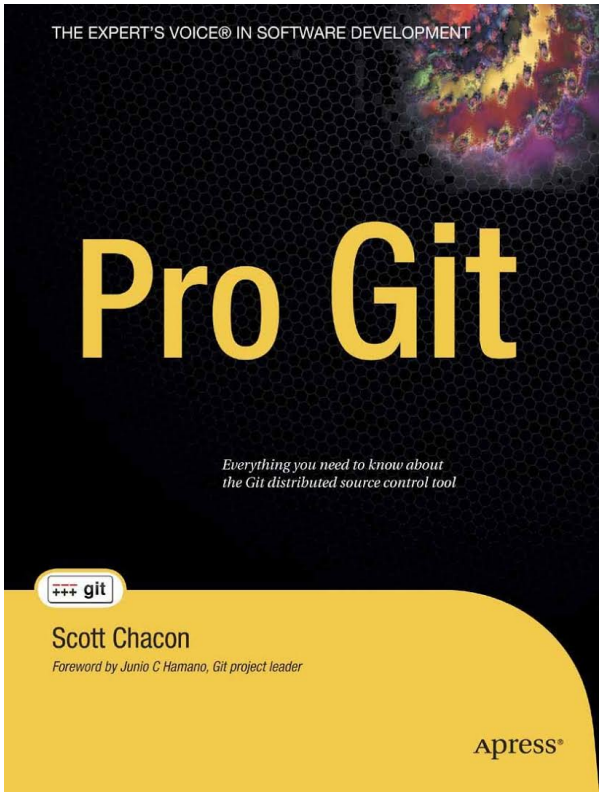
https://github.com/Blunk-electronic

Literatur

http://gitbu.ch/

https://git-scm.com/book/de/v2

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 !