

IT-Infrastruktur

WS 2014/15

Hans-Georg Eßer
Dipl.-Math., Dipl.-Inform.

Foliensatz B':

v1.0, 2014/12/08

- PC als Arbeitsplatz, Teil 2

Vorlesungsübersicht

Seminar

Wiss. Arbeiten

Datenformate und Wandlung

PC als Arbeitsplatz

Ergonomie und Arbeitsschutz

Rechnerstrukturen

(Telekommunikation)

Infrastruktur-Technologie

Zentrale / verteilte IT-Infrastrukturen

(restliche
Mercurial-
Folien)

Folien B'

- Weiterentwicklung von Code in mehreren Zweigen
- Zweige sollen zusammengeführt werden
 - **Merge-Operation**
 - Änderungen eines Zweigs in anderen Zweig übernehmen

Rev. 0:

```
int main () {
  // Block 1
  int i = 0;
  for (; i < 10; i++) {
    printf (i);
  }

  // Block 2
  char c = 'a';
  for (; c <= 'z'; c++) {
    printf (c);
  }
}
```

Rev. 1:

```
int main () {
  // Block 1
  int i = 0;
  for (; i < 10; i++) {
    printf ("%d\n", i); // format code fehlt!
  }

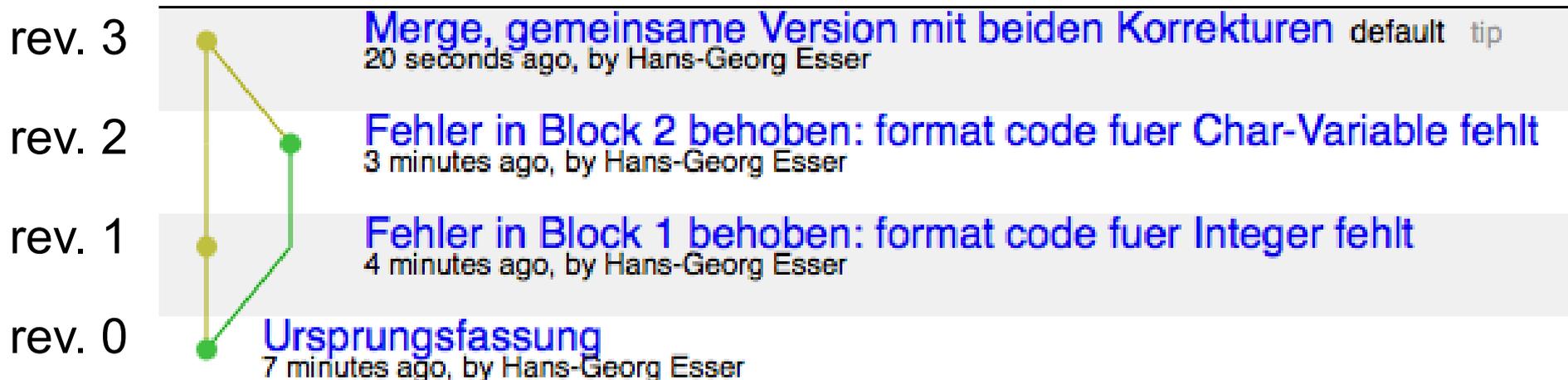
  // Block 2
  char c = 'a';
  for (; c <= 'z'; c++) {
    printf (c);
  }
}
```

Rev. 2:

```
int main () {
  // Block 1
  int i = 0;
  for (; i < 10; i++) {
    printf (i);
  }

  // Block 2
  char c = 'a';
  for (; c <= 'z'; c++) {
    printf ("%c \n", c); // %c!
  }
}
```

```
$ hg merge -r1           # arbeite gerade mit rev. 2
Führe code.c zusammen
0 Dateien aktualisiert, 1 Dateien zusammengeführt, 0
Dateien entfernt, 0 Dateien ungelöst
(Zusammenführen von Zweigen, vergiss nicht 'hg commit'
auszuführen)
$ hg commit
```



- Ergebnis:

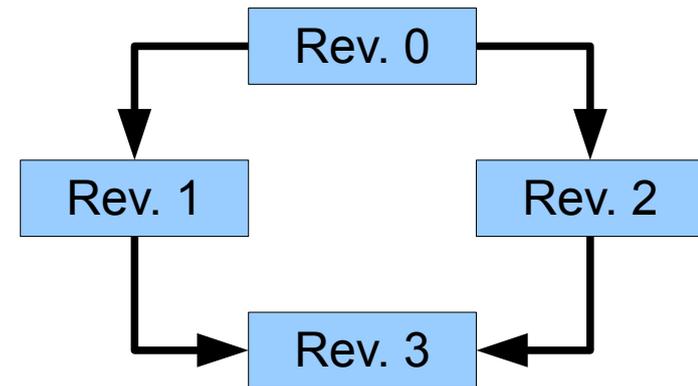
Rev. 3:

```
int main () {
    // Block 1
    int i = 0;
    for (; i < 10; i++) {
        printf ("%d\n", i); // format code fehlt!
    }

    // Block 2
    char c = 'a';
    for (; c <= 'z'; c++) {
        printf ("%c \n", c); // %c!
    }
}
```

Änderung
aus Rev. 1

Änderung
aus Rev. 2

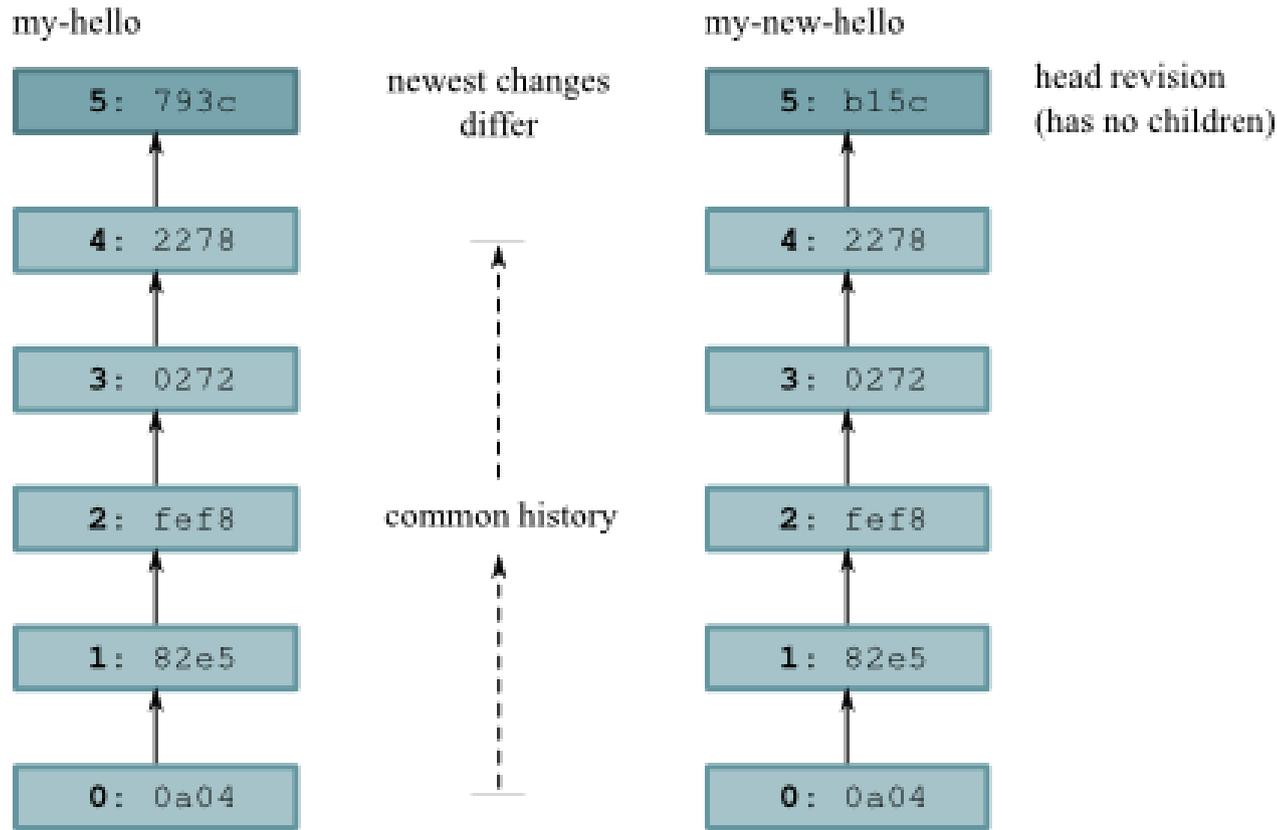


- Merge-Operation erfolgreich, wenn Änderungen in Zweigen verschiedene Bereiche der Datei (oder versch. Dateien) betreffen
- Bei Änderungen derselben Bereiche Konflikt
→ kein automatischer Merge möglich
- Lösen der Konflikte über Tools wie KDiff3
→ KDiff3-Demo

- Klonen (hg clone) erzeugt unabhängige Kopie eines Repositorys
- Bsp.:

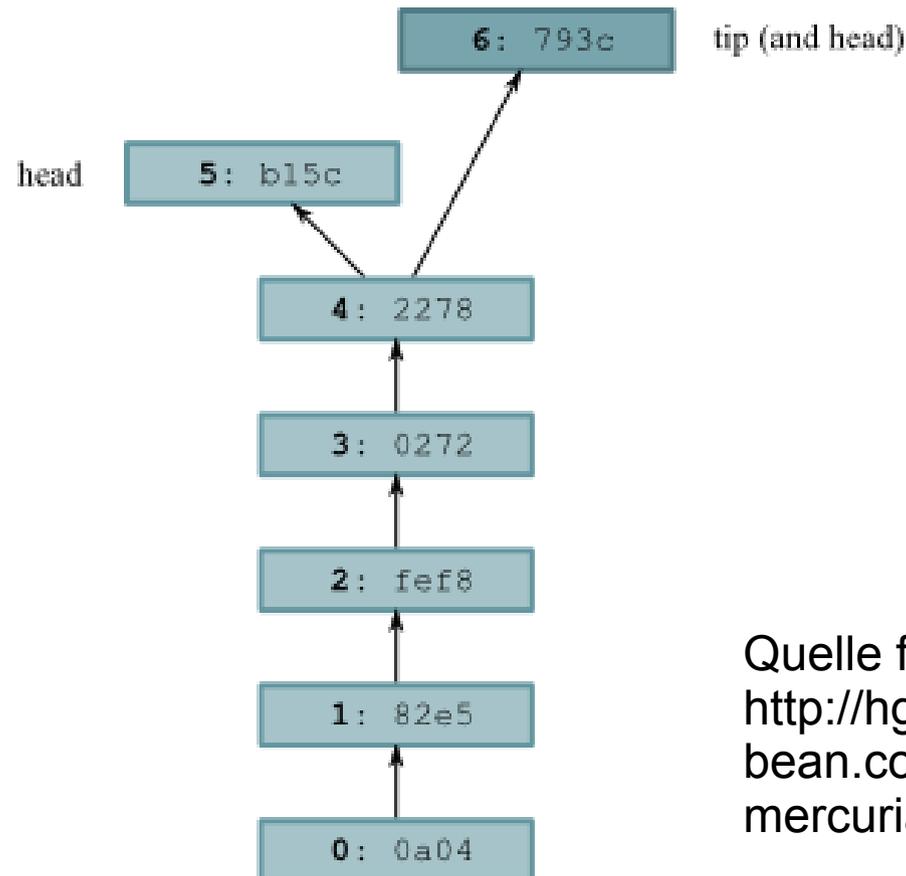
```
$ hg clone \
  my-hello \
  my-new-hello
```

Quelle für Beispiel und Bilder:
<http://hgbook.red-bean.com/read/a-tour-of-mercurial-merging-work.html>



- Pull-Operation (`hg pull`) zieht aktuelle Fassung aus anderem Repository
- Bsp.:

```
$ hg pull \
  ../my-hello
```



Quelle für Beispiel und Bilder:
<http://hgbook.red-bean.com/read/a-tour-of-mercurial-merging-work.html>

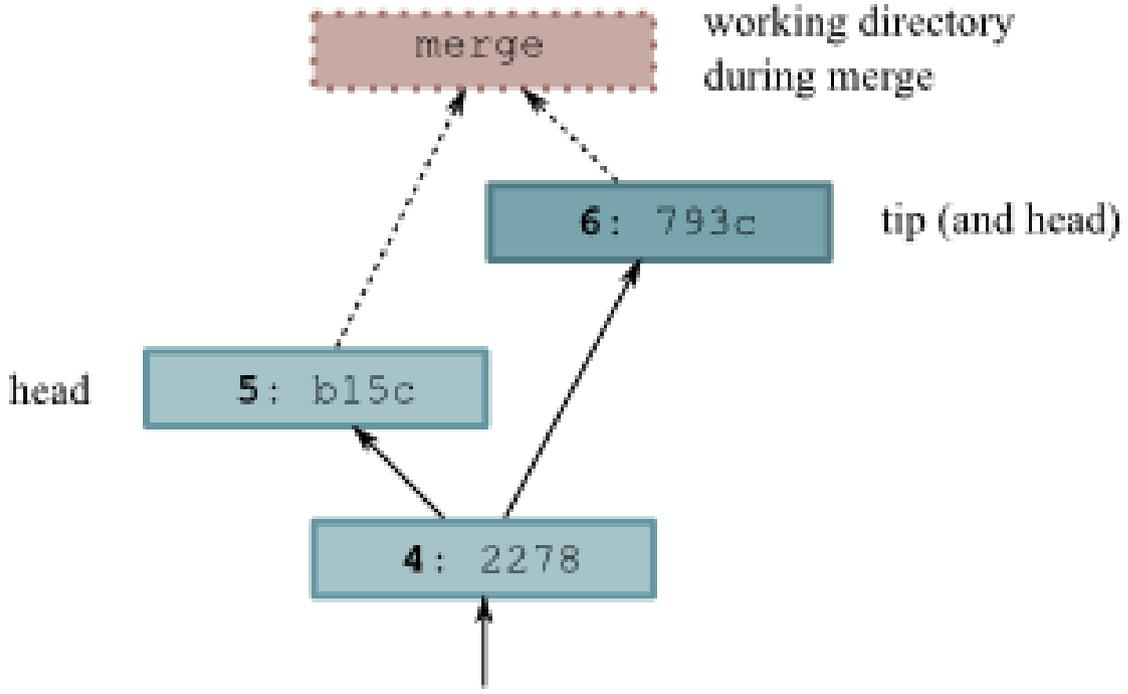
- Zwei Heads (Revisionen ohne Kinder)

```
$ hg heads
changeset: 6:b6fed4f21233
tag:       tip
parent:    4:2278160e78d4
user:      Bryan O'Sullivan <bos@serpentine.com>
date:      Tue May 05 06:55:53 2009 +0000
summary:   Added an extra line of output
changeset: 5:5218ee8aecf3
user:      Bryan O'Sullivan <bos@serpentine.com>
date:      Tue May 05 06:55:55 2009 +0000
summary:   A new hello for a new day.
```

- hg merge vereinigt die beiden heads

- hg merge und hg commit

Working directory during merge



Repository after merge committed

