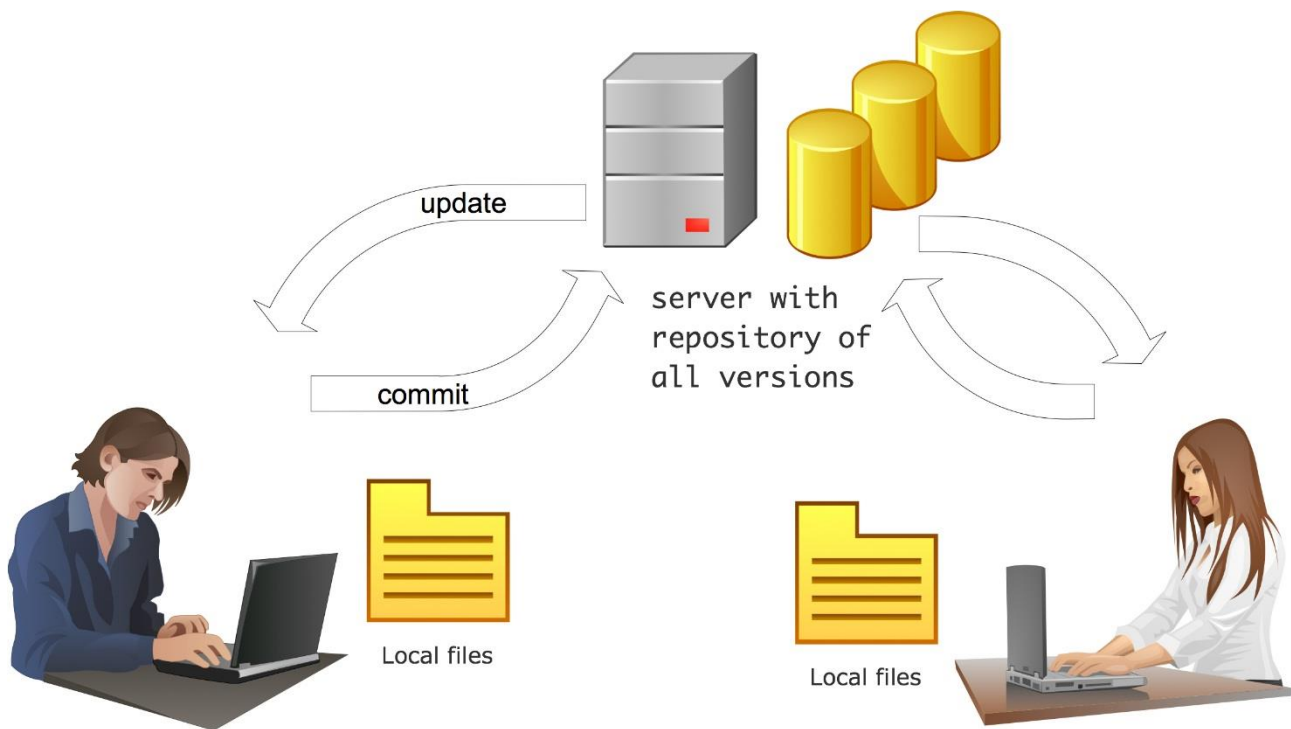


## Dagens program

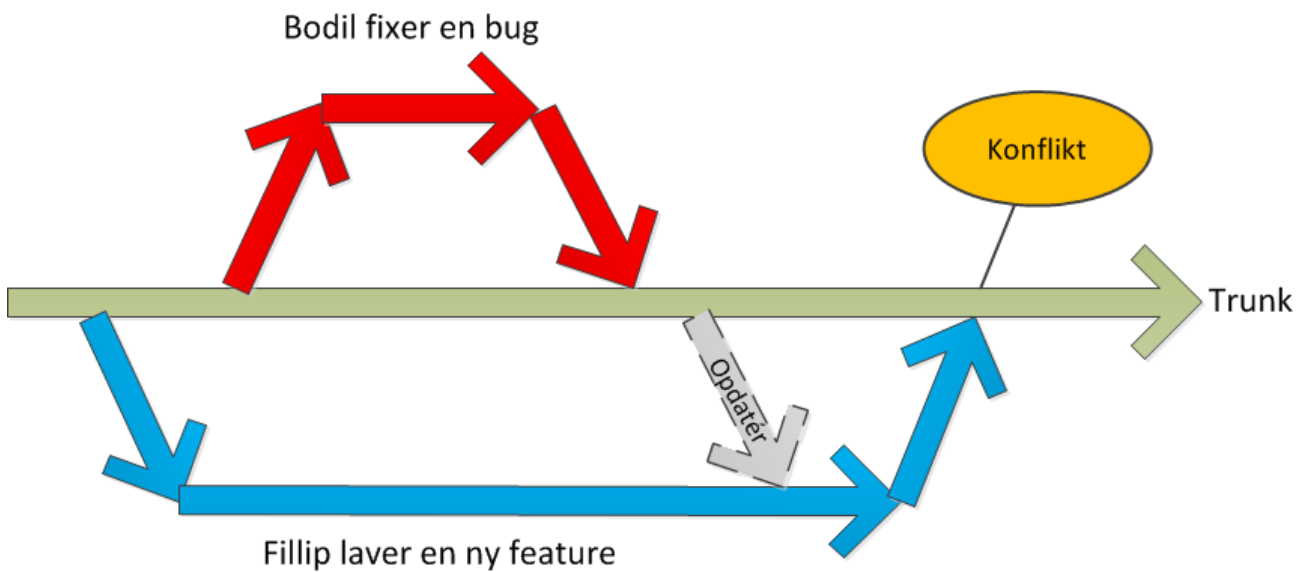
- Opfølgning på sidste opgave
- Versionsstyring på Github
- Opgave: JUnit test af terning. Læg på Github

## Versionsstyring

- Hold sammen på alle filer i et projekt
- Hold styr på versionsnumre hver gang der er ændret noget
- Rul uheldige ændringer tilbage
- Sammenlign versioner og se nøjagtigt hvilke linjer der er ændret
- Organisér samarbejde mellem alle der arbejder på samme projekt
- Lav afgreninger til eksperimentelle formål, forskellige brugere, etc.



Ofte har man et *repository* på en central server



### Centralt versionsstyrings system, f.eks. SVN:

Man kan låse en fil så ingen andre kan ændre den.  
Bodil kan ikke rette fejlen før Phillip er færdig.

### Distribueret versionsstyrings system, f.eks. GIT:

Flere kan arbejde på hver sin version.  
Phillip kan opdatere sin lokale version med Bodils  
rettelse, eller han kan flette sine og Bodils ændringer  
sammen (merge).

## **Fordele ved centralt system:**

- Der er kun én version der er den rigtige

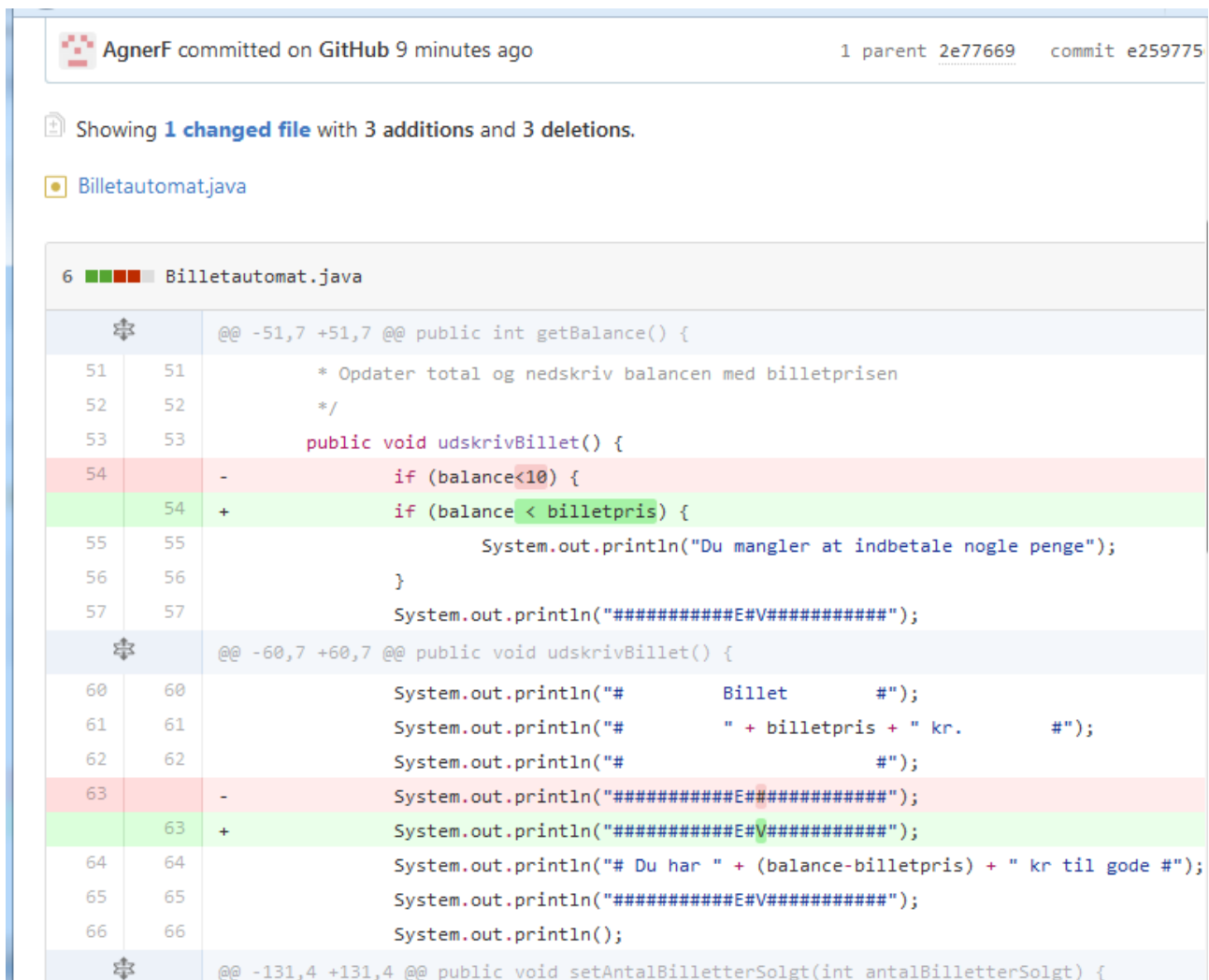
## **Fordele ved distribueret system:**

- Flere kan arbejde samtidigt
- Hver medarbejder kan arbejde på en lokal kopi uden hele tiden at skulle tilgå en central server
- Medarbejdere kan kommunikere med hinanden uden en central server. (Bodil kan sende sin rettelse direkte til Phillip og systemet kan se at Bodils ændring er inkorporeret i Phillips ændring).
- Man kan lave forgreninger f.eks. til eksperimentelle projekter

## Undgå merge konflikter:

- Del op i mindre filer
- Low coupling, high cohesion

## Diff funktionen viser hvad der er ændret:



The screenshot shows a GitHub commit diff for the file `Billetautomat.java`. The commit was made by AgnerF 9 minutes ago, with 1 parent (2e77669) and commit e259775. It shows 1 changed file with 3 additions and 3 deletions.

The diff highlights changes in the `getBalance()` and `udskrivBillet()` methods. In `getBalance()`, line 54 is changed from `if (balance < 10) {` to `if (balance < billetpris) {`. In `udskrivBillet()`, line 63 is changed from `System.out.println("#####E#####");` to `System.out.println("#####E#V#####");`.

```
6 Billetautomat.java
@@ -51,7 +51,7 @@ public int getBalance() {
    * Opdater total og nedskriv balancen med billetprisen
    */
    public void udskrivBillet() {
-       if (balance < 10) {
+       if (balance < billetpris) {
        System.out.println("Du mangler at indbetale nogle penge");
    }
    System.out.println("#####E#V#####");
@@ -60,7 +60,7 @@ public void udskrivBillet() {
    System.out.println("#      Billet      #");
    System.out.println("#      " + billetpris + " kr.      #");
    System.out.println("#      #");
-   System.out.println("#####E#####");
+   System.out.println("#####E#V#####");
    System.out.println("# Du har " + (balance-billetpris) + " kr til gode #");
    System.out.println("#####E#V#####");
    System.out.println();
@@ -131,4 +131,4 @@ public void setAntalBilletterSolgt(int antalBilletterSolgt) {
```

Diff virker kun ved rene tekstfiler.

## Egnede filtyper:

- Ren tekst: `.txt`
- Kodefiler: `.java`, `.cpp`, `.c#`, `.html`, etc.
- Simpel formattering: [markdown](#) `.md`
- Avanceret formattering: [Latex](#) `.tex`

## Kort introduktion til Github:

<https://guides.github.com/activities/hello-world/>



## Opgave:

Opret jer på [Github.com](https://github.com). Alle gruppens medlemmer skal deltage.

Find programmet [terning](https://github.com/AgnerF/Terning) på Campusnet eller på <https://github.com/AgnerF/Terning>

Opret et repository til terning-programmet.

Tilføj en beskrivende [markdown fil](#).

Lav en JUnit test i Eclipse der tester terningen. Den skal teste at terningen ikke kan give andre værdier end 1-6, samt at alle seks værdier forekommer lige hyppigt indenfor en passende fejlmargin. For eksempel, hvis I kaster terningen 60000 gange skal hver værdi forekomme ca. 10000 gange  $\pm 400$ .

Tilføj JUnit test filerne til jeres repository på github.

Lav en Branch med navnet BugFixes i jeres repository.

Ret de fejl I har fundet i terning-programmet og læg dem i BugFixes vha. Commit.

Lav en Pull Request der anmoder om at indsætte rettelserne i jeres master branch.

Acceptér denne pull request med en Confirm merge.

Kontrollér at ændringerne kan ses: Klik på en ændret fil og tryk History for at se ændringerne, eller Blame for at se hvem der har ændret.

Denne opgave afleveres på Campusnet som et link til jeres repository på github.

Det skal kunne ses af historien på github at alle gruppens medlemmer har lavet et eller andet.