

Versionsstyring

© NIELS ØSTERGAARD 2019. MULTIMEDIEDESIGNERUDDANNELSEN,
ERHVERVSAKADEMI DANIA, SKIVE

Indhold

- ▶ Læringsmål
- ▶ Hvorfor versionsstyring
- ▶ Hvad er versionsstyring – forskellige tilgange
- ▶ Git og GitHub
- ▶ Kort introduktion til de væsentligste begreber
- ▶ Øvelse (Hello World)
- ▶ GitHub Desktop – anvendelse i praksis
- ▶ Opsamling
- ▶ Opgave

Læringsmål

- ▶ Læringsmål i studieordningen er et færdighedsmål: "***anvende centrale teknologier og metoder til versionsstyring i en multimedieproduktion.***"
- ▶ Det kan oversættes til:
- ▶ Efter i dag:
 - ▶ Har du overblik over, og kan forklare de væsentligste begreber
 - ▶ Versionsstyring, Git, repository, branch, commit, pull request, merge
 - ▶ Kan du anvende versionsstyring i praksis på dine webprojekter med anvendelse af GitHub Desktop



Hvorfor versionsstyring?

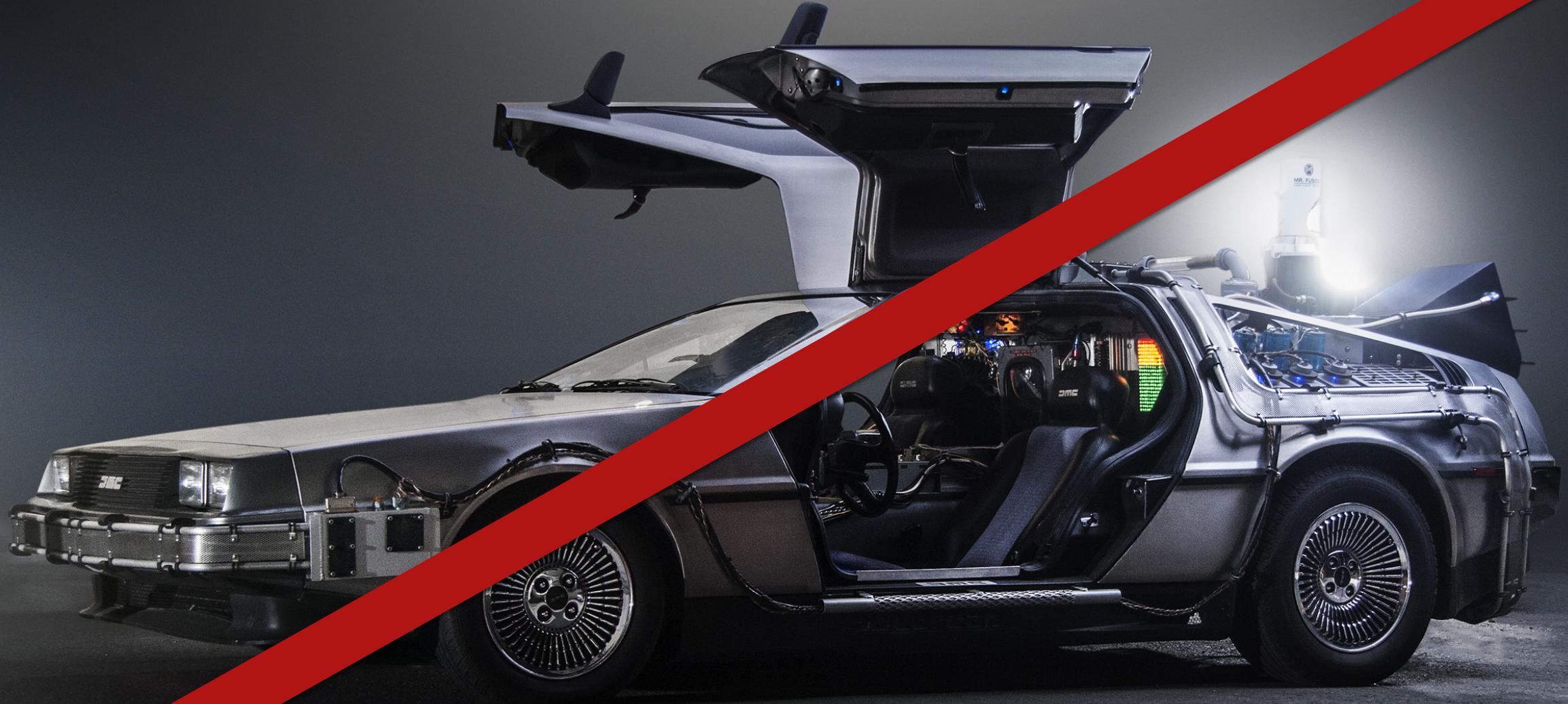
Hvorfor versionsstyring

- ▶ For at undgå fejl
- ▶ For at kunne gå tilbage, hvis de alligevel opstår
- ▶ For at holde styr på flere delprojekter
 - ▶ Samme elementer kan måske laves i forskellige versioner til forskellige kunder
- ▶ For at kunne samarbejde
- ▶ ...

Hvad er versionsstyring?

- ▶ En metodisk tilgang!
- ▶ Kan være så simpelt som navngivningen af filer og mapper
- ▶ Kan måske suppleres med noter om hvad der ændres fra version til version
- ▶ VCS (Version Control System) kan understøtte versionsstyring (og samarbejde)
- ▶ Softwaren holder styr på forskelle mellem versioner
- ▶ Kan enten være en tjeneste i skyen, eller installeret på en lokal server
- ▶ En af de mest udbredte måder blandt webudviklere er baseret på Git-protokollen, en anden er SVN (Subversion)







En slags tidsmaksine - javel

MEN **KUN** TIL AT REJSE **TILBAGE** I TIDEN! – MEN DET KAN OGSÅ VÆRE ET
STORT SPRING FREMAD!

Hvor bruges versionsstyring

- ▶ I alle typer it-projekter
 - ▶ Udvikling af administrative systemer, apps, embeddede systemer, frameworks, libraries og komponenter, websites mm
- ▶ Hvem
 - ▶ "Alle" der arbejder professionelt med it-udvikling
 - ▶ Store organisationer
 - ▶ Iværksættere
 - ▶ Hobby-kodere
 - ▶ Små og store Open Source -projekter
- ▶ I princippet også anden information end kildekode
 - ▶ Aftaler
 - ▶ Rapporter
 - ▶ ...

Git

- ▶ Git er ikke GitHub – det kommer vi til
- ▶ Git er en standard (protokol)
- ▶ Git er et versionsstyringssystem
- ▶ Er oprindeligt skrevet af Linus Thorvalds for, at holde styr på udviklingen af Linux efterhånden, som der kom flere med
- ▶ Du kan selv oprette en Git-server
- ▶ - eller bruge en onlinetjeneste som GitLab, BitBucket, GitHub og mange andre



GitHub

- ▶ Nu er det altså ikke Git..
- ▶ Er en webplatform der tillader at man kan oprette Git-repositories (skyen) - Altså en server, så du ikke selv behøver installere en
- ▶ Har en del flere features indbygget – bla. til samarbejde, projektstyring mv.
- ▶ Er tænkt som et kodedelings-site
- ▶ Et socialt medie
- ▶ Er meget udbredt – "er du ikke på GitHub??"
- ▶ Portfolio for kode – det er efterhånden en udbredt del af ansættelser
- ▶ Kan bruges via kommandolinje eller desktopprogram



GitHub

- ▶ Mange store Open Source –projekter hostes her
- ▶ Et sted at hente store og små kodestumper til egne projekter
- ▶ Mulighed for private repositories (begrænset hvis man ikke betaler)
- ▶ Har en studiepakke – som måske kan være relevant..

- ▶ Bliver stedet hvor dine fremtidige projekter skal have et repository



Men er GitHub den bedste løsning?

- HVILKE FORDELE OG ULEMPER ER DER VED EN HOSTET LØSNING



GitHub er god til at håndtere
samarbejde mellem flere

- MEN VI FOKUSERER I FØRSTE OMGANG PÅ JERES PRIVATE
VERSIONSSTYRING OG IKKE SAMARBEJDE

Væsentlige begreber

- ▶ **Repository** (repo) – en samling af filer, som en projektmappe
- ▶ **Branch** - En forgrening, en parallel version i dit repo, men berører ikke master
- ▶ Master Branch - Hovedforløbet, den der gælder
- ▶ **Commit** - En revision, som at gemme, men med mulighed for at finde tilbage – man kan også kalde det en milepæl
- ▶ **Pull Request** - Et forslag til en ny udgave (rettelser)
- ▶ **Diff** – Difference, en forskel mellem to commits
- ▶ **Merge** - Når ændringer i en branch samles med en anden (eks. gennem en pull request)
- ▶ **Pull** – Når du henter ændringer og merger dem

Øvelse – forstå begreberne

1. Opret en bruger på GitHub (Vigtigt: brug din Dania-mail)
1. Følg Hello-world eksemplet: <https://guides.github.com/activities/hello-world/>
 - a) Opret et repository (husk readme)
 - b) Start en ny branch
 - c) Lav ændringer
 - d) Push dem (Commit)
 - e) Åbn en Pull Request
 - f) Merge ændringer
- g) Efterfølgende læses (5-10 min):
<https://guides.github.com/introduction/flow/>

Hvad fik vi ud af øvelsen?

- ▶ Har vi styr på begreberne?
 - ▶ Repository
 - ▶ Branch/Master Branch
 - ▶ Commit
 - ▶ Pull Request
 - ▶ Merge



Men hvad med filer?

For at tilføje filer til et projekt:

- ▶ Skal du have et værktøj til at styre dine filer og ændringerne (Commit)
- ▶ To(+) muligheder med GitHub:
 - ▶ Brug GitHub Desktop
 - ▶ Brug kommandolinjefunktioner
 - ▶ (eller brug programmer der har indbygget understøttelse af Git)



Opret et webprojekt med GitHub Desktop

(SE EVT. [HTTPS://HELP.GITHUB.COM/DESKTOP/GUIDES/](https://help.github.com/desktop/guides/) FOR HJÆLP)

GitHub Desktop

- ▶ Installer programmet
- ▶ Tilføj din bruger
- ▶ Opret et repository
 - ▶ Placer det hensigtsmæssigt, så du kan finde det
- ▶ Opret HTML og CSS (kopier evt. et par filer fra tidligere)
- ▶ Commit undervejs – gerne ofte og tidligt!
- ▶ Når du har en side der ser ok ud, så lav en branch
- ▶ Lav tydelige ændringer i siden
- ▶ Lav en pull request
- ▶ Kommenter requesten
- ▶ Lav en merge (uden at slette branch)
- ▶ Lav en release

Litteratur

- ▶ Væsentlig
 - ▶ <https://guides.github.com/activities/hello-world/>
 - ▶ <https://guides.github.com/introduction/flow/>
 - ▶ <https://desktop.github.com/>
 - ▶ <https://help.github.com/desktop/guides/getting-started-with-github-desktop/>
 - ▶ <https://help.github.com/articles/github-glossary/>
- ▶ Supplerende
 - ▶ <https://sethrobertson.github.io/GitBestPractices/#commit>
 - ▶ <https://www.codefellows.org/blog/git-and-github-what-s-the-difference/>
 - ▶ <https://www.git-tower.com/blog/git-hosting-services-compared/>