git introduktion
Del 1 av 4

https://github.com/Rnqst

Roland Ronquist

Git-kursen

- 15 minuter rast varje timme
- Ställ frågor!
- Det är säkert fler som undrar över samma sak

git-kursen schema

- 9:00 – 9:45
- Rast
- 10:00 – 10:45
- Rast
- 11:00 – 12:00

git SCM – Source Code Management

- Vi kommer att lära oss att använda git
- Du blir effektivare som programmerare
- Det fungerar bättre att arbeta i grupp
- git förbättrar kommunikationen i laget
- git effektiviserar kundsupporten

git SCM – Source Code Management

- Versionshantering / revisionshantering
- Håller reda på hur koden utvecklas över tiden
- Vad levererades till vem? Närdå?
- Sparar plats genom att bara lagra förändringar

Hur arbetar en programmerare?

- Bygger program från text
- Testar nya idéer
- Jämför olika versioner
- Sammarbetar

Vad gör en ”vanlig” kontorsarbetare?

- Arbetar mycket i program som egentligen inte genererar data lokalt på maskinen
- Använder program med komplexa filstrukturer
- Skapar dokument som blandar många typer av information

Programmerarens behov

- Hanterar många textfiler
- Det är viktigt att veta hur koden ändrats över tiden
- Stöd för att kopiera moduler mellan olika versioner
- Det skall vara enkelt att prova saker
- Säkerhetskopiering
- Sammarbete

Vad gör ett versionshanteringssystem?

- Programmet hanterar ”projekt” med många filer och kataloger
- När koden checkats in i systemet går det att markera olika versioner
- Somliga föredrar ordet revision framför version

Vad gör ett versionshanteringssystem?

- Håller både reda på grundversioner och alla specialversioner
- Det går att knoppa av huvudlinjen till en ny gren
- Grenarna kan integreras tillbaka till huvudlinjen

Vad gör ett versionshanteringssystem?

- När programmeraren checkar in kod i systemet hamnar den på en server som kan säkerhetskopieras
- Git hjälper även till med kopiering av kod mellan olika grenar
- Versionshanteringssystem spar plats genom att bara lagra skillnaden mellan olika versioner

Vad gör ett versionshanteringssystem?

Termer

- Master, main, trunk, stam, rot
- Gren, branch, fork….
- Revision, version

Bilden på föregående sida från:

https://git-scm.com/about

Varför är git så bra?

- Varje användare har en helt egen kopia av koden
- Alla kopior kan innehålla alla versioner
- Alla kopior kan vara jämlika, en server är inte nödvändig

Varför är git så bra?

- När du delar ett projekt till någon kan du välja att dela allt eller bara delar
- Exempelvis kan du dela en specifik gren
- Eller så kan du dela koden som den ser ut just nu utan historia

Varför är git så bra?

- git tvingar dig inte att ha en server
- Börja utveckla något själv på din dator, du har tillgång till alla verktyg och funktioner
- När du får fler som vill hjälpa till kan ni enkelt dela koden

Varför är git så bra?

- Jämfört med de flesta andra system är git väldigt snabb
- När två grenar skall slås ihop får du hjälp av git
- Det finns många tjänster på nätet med git

Varför är git så bra?

- Det är osannolikt att git saknar någon funktion
- Dessutom är det idag det populäraste systemet
- Det är stabilt och säkert
- Det finns på de flesta plattformar

Hur går vi vidare?

- Läs boken Pro Git som du antingen kan beställa som en tryckt bok eller ladda hem
- https://git-scm.com/book/en/v2
- Den finns översatt till 30-talet språk
- Dock är inte alla översättningar klara

Github

- Github är världens populäraste att dela källkod på nätet
- Github är mer än bara git, det finns en rad ”sociala” funktioner
- Men även praktiska saker som användarforum och felrapporteringssystem

Github

- https://github.com/
  - Registrera ett eget konto på github
  - De behöver en fungerande epostadress för registreringen

Github

- Registrera ett eget konto på github
- De behöver en fungerande epostadress för registreringen

Hur används git?

- Från början styrde alla git via kommandon
- Nu är det vanligt att git används från en utvecklingsmiljö eller texteditor
- Exempelvis: Eclipse, Visual Studio, Sublime Text

Hur används git?

För att göra det enkelt att komma igång finns även fristående verktyg

Installera Github Desktop

https://desktop.github.com/

Då behöver du ett konto på github.

Github Desktop

- Vi kommer att utgå från github desktop i våra övningar
- Det är ändå några begrepp och funktioner du måste känna till

Blandade tankar om git

- För 30-40 år sedan gjordes nästan alla program från en terminal eller ett terminalfönster
- Det arbetssättet är ofta väldigt effektivt även idag
- Mån programmerare trivs i integrerade utvecklingsmiljöer

Blandade tankar om git

- Även om du inte tänker köra några kommandon så är de bra som utgångspunkt för inlärningen
- De heter ”vad de gör”
- Ni hittar lätt dokumentation som förklarar funktionen

Github desktop

- Github desktop är en bra startpunkt
- Du behöver inte använda kommandon
- Du slipper låsa upp dig till en viss utvecklingsmiljö

Github desktop

- https://docs.github.com/en/desktop/installing-and-configuring-github-desktop/overview/getting-started-with-github-desktop
- https://www.youtube.com/githubguides
- https://www.youtube.com/watch?v=SWYqp7iY_Tc

Inför morgondagen

Tänk på följande frågor:

Din gamla dator gick sönder men du hade checkat in all din kod på github innan. Har du då tappat information när du tittar på koden i din nya dator efter att ha checkat ut den där igen?

Inför morgondagen

Om vi fortsätter frågan med att även github kraschar efter att du lyckats hämta ut koden till din nya dator. Vad finns det för möjlighet att då kopiera din kod till dina kollegor om github ligger nere?

Om det går att kopiera till kollegorna, blir deras kopior då lika kompletta som om de tankat sakerna direkt från github?

Inför morgondagen

- Varför lägger så många upp sina program på github?
- Är det inte bättre att sälja programmen och bli rik?