Git-Workshop

- Organisation
- Material
- Orga-Checkliste
- Skript zu den Live-Demos des Workshops

Organisation

Übersicht

Git-Workhop zur Vorbereitung auf das Gruppenprojekt in der FOP gegen Ende des Wintersemesters.

Wann?

Ende des Wintersemesters; bisher immer im Februar. Möglichst nicht zu weit nach Beginn des Projekts, da sonst Interesse verloren geht ("Wir haben bereits angefangen, da steigen wir nicht mehr auf Git um!"). Jedoch auch nicht zu früh, da sonst durch andere Abgaben und Verpflichtungen keine Zeit besteht.
Die Übung wurde bisher immer im Anschluss an den Vortrag veranstaltet.

Zu berücksichtigen:

- Termine des FOP-Projekts: Start, Ende, HDA-Vortragstraining
- Andere Termine: Abschlussvorlesung des Mentorensystems, Vorlesungen, Klausuren
- Uhrzeit Übung: Evtl. Mittagspause zwischen Vortrag und Übung

Dauer (Erfahrungswerte):

- Vortrag: ca. 60 Minuten (inkl. Live-Demos)
- Übung: nach 1,5h bis 2h löst es sich auf

Vergangangene Termine:

- 2020: 20.02. ca 60 Leute insgesamt beim Vortrag, ca 8 beim Workshop, FOP fast alle, nicht Info ~7 (WInf, Mathe, CogSci), Git Erfahrung ~5
- 2019: 27.02.: Geringe Teilnehmerzahl
- 2019: Vortrag ca. 40-60 Personen; Übung: max. 10 Personen
- 2018: 21.02.: ?
- 2017: 16.02.: ?
- 2016: 12.02. (Im Anschluss an Abschluss des Mentorensystems): Sehr hohe Teilnehmerzahl, keine Übung veranstaltet

Wer?

Orga

- 1-2 Menschen für Vortrag und Organisation
- Zusätzliche Helfer für Übung je nach Teilnehmerzahlen: i.d.R. 2-3 Menschen

Teilnehmer

- Studierende der FOP: B.Sc. Informatik, LaG Informatik, Winf, Nebenfach Informatik, Studiengänge mit Informatik-Anteil (PsychIT, CE, IST)
- Andere Interessierte: ETIT

Wo?

Geeignete Hörsäle:

- S2|02 C205 (Großer Informatik-Hörsaal)
- S1|05 122 (Georg-Wickop-Hörsaal)
- Vielleicht Hexagon je nachdem wann es fertig wird und wie es dann aussieht

Anforderungen an Hörsäle:

- Min. 2 Beamer
- Min. 2 separate Inputs für verschiedene Laptops
- Mikrofone

Übung:

- Großer C-Pool (S2|02 C005)

Inhalte

Vortrag

- Grundlagen von Git
- Live-Demo ohne Remotes
- Live-Demo mit Remotes

Live-Demo mit Remotes

Am besten gehalten durch zwei Vortragende mit zwei Laptops, die die Rollen von Alice und Bob übernehmen. Als Beispielplattform kommt OpenProject des FB Informatik zum Einsatz.

Was wird gezeigt?:

- Erstellung eines Projekt (siehe ISP Wiki für detaillierte Informationen)
- Einrichtung SSH-Key
- Hinzufügen von anderen Teilnehmern
- Ablauf aus Präsentation: Basic, Auto-Merge, Manual-Merge

Hinweise für zwei Vortragende:

- Am besten Linux + Windows als Systeme zeigen
- Unterschiede deutlich machen: SSH-Key-Generierung, Git-Shell

Übung

- Das im Vortrag gezeigte Ausprobieren
- Bearbeitung des Übungsblatts
- Für Orga: Bereitstellung der Lösung online

Feedback

Wenn FOP-Dozenten Feedback zu Projekt einholen, zusätzliche Fragen:

- Kenntnis vom Git-Workshop-Angebot der Fachschaft Informatik? (j/n)
- Teilnahme am Git-Workshop? (j/n)
- Feedback/Anmerkungen (Freitext)

Nice-to-Have TODOs

- Bebilderte Anleitung zur Git-Installation auf Windows
- Bebilderte Anleitung zur Erstellung eines SSH-Keys auf Windows (vielleicht hat das ISP Wiki so etwas?)

Material

Vorlesung + Vortrag + Aushang

Siehe D120 GitLab.

Werbung

Öffentliche Ankündigung als E-Mail

An alle Fachschaften mit relevanten Studiengängen:

- wir@d120.de
- fachschaft@fs1.de
- fachschaft@ce.tu-darmstadt.de
- info@fs-ist.de
- fachschaft@mathematik.tu-darmstadt.de
- fachschaftlag@lists.tu-darmstadt.de
- fachschaft@physik.tu-darmstadt.de
- mail@fspsy.de
- fachschaft@fs-etit.de

> Liebe Fachschaften,
>
> die Fachschaft Informatik plant zum Start des Programmierprojekts in Funktionale und objektorientierte Programmierkonzepte (FOP, ehemals GDI 1) einen Workshop zum Versionskontrollsystem Git [0] zu veranstalten. Er steht allen Interessierten offen.
>
> Inhaltlich werden wir die Motivation und Grundlagen von Versionskontrollsystemen anhand von Beispielen vermitteln. Dies umfasst die Nutzung von Git-Grundbefehlen über die Shell und die Zusammenarbeit über das ISP-SCM [1], welches auch im Rahmen des FOP-Projekts empfohlen wird.

> Starten werden wir mit einem Vortrag, in dem Versionskontrollsysteme allgemein und Git als konkretes Beispiel präsentiert werden. Im Anschluss wird eine Übung angeboten, in welcher Git ausprobiert werden kann. Diese Übung wird durch erfahrene Studierende betreut, sodass Fragen gezielt beantwortet werden können.
>
> Der Vortrag findet am *Mittwoch, den 27. Februar 2019 ab 12:00 in S105/122 (Altes Maschinenhaus)* statt. Nach einem Vortrag von ca. 60 Minuten wird die *Übung in S202/C005* stattfinden.
>
> Für weitere Fragen und Anregungen stehen wir natürlich gerne zur Verfügung.
>
> [0] https://git-scm.com
> [1] https://scm.informatik.tu-darmstadt.de

Andere Platformen

- FOP-Dozenten um Werbung im Kurs bitten
- Post auf dasWesentliche
- Teilen von Post auf Facebook und Twitter
- Eventuell Aushang

Orga-Checkliste

- Termin festlegen und Raumbuchung: 1,5 bis 2 Monate vorher
  - Abstimmung mit Dozenten über Termin bzgl. anderer FOP-Termine
  - Buchung des Hörsaals (siehe allgemeine Raumbuchung)
  - **KEINE** Buchung des C-Pools (Wir wollen den Studierenden keine Lernräume wegnehmen)
- Werbung: 1,0 bis 1 Monat vorher
  - Hinweis an Dozenten
  - Rundmail an Fachschaften (siehe Material)
- Anwerben von Helfern für Übung: 2 bis 4 Wochen vorher
  - Eine Bitte über fs@ senden reicht meistens
- Erinnerungs-Werbung: 1 Woche vorher
  - Sociel Media Kanäle o.ä.
- Tag des Vortrags:
  - Transponder für Tote Briefkästen organisieren (entweder Hörsaal-Transponder der Fachschaft wenn Berechtigung vorhanden, Hausmeister, Pforte im Alten Hauptgebäude, ...)
  - Paar Exemplare gedruckter Übungen bereitlegen im C-Pool
  - Vortrag
- Nachbereitung:
  - Rückgabe des Transponders
  - Dokumentation aktualisieren (v.a. Teilnehmerzahlen!)
  - Hochladen der Lösungen zu den Übungen
  - Feedback über FOP-Projekt einholen

Dazu natürlich noch eine Überarbeitung des Workshopmaterials, falls erforderlich.

Skript zu den Live-Demos des Workshops

1. Demo - Git alone

Vorbereitung

- git-scm.org
- Git für Windows herunterladen, installieren (auf Editor-Auswahl hinweisen, ansonsten Standardeinstellungen)
- Musterordner öffnen
- Shell öffnen in Musterordner
- git config --global user.name Alice
- git config --global user.email alice@d120.de
- git init

Dateien hinzufügen und Commits erstellen

- git status
- git add example.java
- git status
- git commit -m "Initial commit"
- git status
- In example.java add implementieren
- git add example.java
- git commit Implemented add Solved task 1
- In example.java sub implementieren
- git add example.java
- git commit -m "Implemented sub"
- git log --oneline --graph

SSH-Key generieren

- ssh-keygen
- notepad .ssh/id_rsa.pub - copy
- github.com, Settings, SSH and GPG keys, New SSH key - paste

Neues Repo anlegen und einrichten

- Create new repository, Private
- copy-paste commands from site to configure remote
- git push

2. Demo - Git together

Collaborators einladen

- Settings, Manage Access, Invite a collaborator
- [B] Einladung annehmen (Link per Mail)
- [B] git clone

Automatic merge

- [A] editiert
- [B] editiert andere Zeile
- [A] pusht
- [B] pusht, bekommt Fehler, pullt + pusht und es geht automatisch

3. Demo - Git in conflict

Merge conflict

- [A] editiert
- [B] editiert gleiche Zeile
- [A] pusht
- [B] pusht, bekommt Fehler, pullt + pusht und bekommt die Fehlermeldung "Automatic merge failed".
- [B] löst den Konflikt

4. Demo - Branches

Branch erstellen und wieder mergen

- git checkout -b feature/multiply
- example.java bearbeiten
- git add example.java
- git commit -m "implement multiplication"
- git log --oneline --graph
- git checkout master
- git merge feature/multiply