Die Projekttools

Files, Git, Tickets & Time

Agenda

- Die Abgabe von Dokumenten: Files
- Das Pflegen von Software: Versionskontrolle mit Git
- Management von Anforderungen: Tickets
- Management von Zeit: Time

Files Tool

Dateiverwaltung

Files Tool

- Abgabe von erstellten Dokumenten
- Versionskontrolle von Dateien
- Teilen von Dateien (File Sharing)
- Einfügen von google docs

Git

Die Versionierungssoftware

Git

- clone/init
- add & remove
- commit
- branches
- push
- fetch/pull
- merge

git clone/init

- Klonen eines bestehenden repositories
  - git **clone** *url*
    - *url* gibt die Adresse des Servers an, auf dem das repository zu finden ist
- Erstellen eines neuen lokalen repository im aktuellen Verzeichnis
  - git *init*

git add/rm

- Hinzufügen einer Datei zum repository, damit diese versioniert werden kann
  - git **add** *file1 file 2*
    - es werden die Dateien *file1 file2* hinzugefügt
  - git **add** *dir*
    - es wird der angegebene Ordner *dir* hinzugefügt
  - git **add** .
    - es werden alle Dateien (rekursiv inkl. aller Unterordnern) hinzugefügt
  - git **rm** *file1 file2*
    - es werden die Dateinen *file1 file2* entfernt

git commit

- Änderungen werden in das repository eingefügt
  - git **commit** *file1 file 2* [-m *msg*]
    - neue Versionen der Dateien *file1 file2* werden hinzugefügt.
    - *msg* ist eine Nachricht zur den Änderungen (optional)
    - ohne *msg* öffnet sich automatisch ein verfügbarer Editor für die Nachricht (bspw. vim)
  - git **commit** –a [-m *msg*]
    - Alle Änderungen an dem repository bekannten Dateien werden hinzugefügt

git branch

- Alle branches anzeigen lassen
  - git **branch**
- Neuen branch erstellen
  - git **branch** *new-branch*
    - basierend auf dem aktuellen branch
  - git **branch** *new-branch rev*
    - *basierend auf der Revision* rev
  - git **branch** --track *new-branch remote/remote-branch*
    - Basierend auf einem entfernten branch *remote-branch* auf dem *Server remote*

git checkout

- Zwischen verschiedenen branches wechseln
  - git **checkout** *branch*
    - wechselt zum branch *branch*
  - *git* **checkout** –b *new-branch*
    - erstellt einen neuen branch *new-branch* basierend auf dem aktuellen und wechselt zu diesem
  - git **checkout** *rev*
    - wechselt zur Revision *rev* ohne einen neuen branch zu erstellen
  - git **checkout** -b *new-branch rev*
    - erstellt einen neuen branch *new-branch* basierend auf der Revision *rev* und wechselt zu diesem

git branch delete

- Löschen eines branches
  - git **branch** –d *branch*
    - *löscht einen branch im lokalen repository*
  - git **push** *remote* :heads/*branch*
    - löscht einen branch auf dem Server *remote*

git push

- Alle Änderungen am lokalen repository werden auf den Server geladen
  - git **push** [remote]
    - lädt alle Änderungen auf den entfernten Server *remote*
  - git **push** *remote branch*
    - lädt alle Änderungen am *branch* auf den entfernten Server *remote*

git fetch/pull

- Laden des repository oder der neusten Änderungen vom Server in ein lokales repository
  - git **fetch** *remote*
    - Lädt die aktuellen Änderungen vom Server *remote* herunter, führt aber keinen merge im lokalen repository durch sondern erstellt einen neuen branch (default: *remote*/master).
  - git **pull** *remote*
    - Es werden alle aktuellen Änderungen vom Server *remote* geladen und ein merge mit dem lokalen repository gemacht (default branch: master).

git merge

- Zusammenführen von branches
  - git **merge** *branch*
    - *branch* wird in den aktuellen branch (bspw. master) geladen
  - git **merge** *branch* –no-commit
    - merge ohne commit, so kann man das Ergebnis des merges angucken, verbessern und später ein commit machen

git: links

- http://git-scm.com/downloads
  - Download git
- http://code.google.com/p/tortoisegit/
  - Windows Interface für git
- http://gitref.org/
  - Referenz zum Nachschlagen mit weiteren Befehlen

git & assembla

- ssh key erstellen
  - Puttygen auf windows
  - **ssh-keygen** -t dsa
  - **ssh-keygen** –t rsa
- ssh public key auf assembla hochladen

Tickets

Organisieren von Aufgaben

Tickets

- Tickets neu erstellen
- Einem Ticket folgen
- Ein Ticket kommentieren
- Tickets abschließen
- Milestones erstellen
  - Aktuelle milestones
  - Zukünftige milestones
  - Backlog
- Stories
  - Zusammenfassen von milestones zu stories

Time

Zeitmanagement

Time

- Angabe von an einem Ticket gearbeiteter Zeit
- Schätzung, wie viel Zeit noch erforderlich ist für ein Ticket

Fragen?