Der Transparente Mensch: Die unvermeidliche Preisgabe von Metadaten

Arne Beer, MN 6489196, University of Hamburg

20.08.2019

1 Abstract

Metadaten spielen eine bedeutende Rolle in der Welt der Datenanalyse. Simple Informationen über einen Menschen, wie zum Beispiel der Standort verknüpft mit Zeitpunkten, verschiedene Aktivitäten oder Vorlieben, liefern enormes Potential, um diesen zu analysieren und kategorisieren. In vielen Fällen wird aktives Sammeln von Daten, Data Mining und Big Data Analysis betrieben und Plattformen wie Facebook, haben sich voll diesem Ziel verschrieben.

Metadaten sind jedoch tückisch und befinden sich an viel mehr Orten als man vielleicht auf den ersten Blick vermutet.

Dieses Paper wird sich speziell mit der Intransparenz in Bezug auf die Freigabe von Metadaten bei bestimmten Tools auseinandersetzen, welche eigentlich nicht für diesen Zweck vorgesehen sind. Speziell betrachten wir in diesem Falle das Tool *Git*, welches hauptsächlich zur Versionskontrolle von Quellcode in Informationstechnischen Projekten verwendet wird. Zudem wird die populäre Website *GitHub* betrachtet, welche als Open-Source Plattform dient, auf der jeder Entwickler seine Projekte öffentlich zur Verfügung stellen kann.

2 Einleitung

Das Ziel dieser Arbeit ist, die Notwendigkeit von Datenerfassung zu betrachten. In einigen Umfeldern, speziell im Informationstechnischem Bereich, ist die Erfassung von Metadaten unvermeidlich und teilweise unabdingbar. Daten werden zur Analyse von technischen Prozessen benötigt oder um z.B. die Verantwortlichkeit eines Entwicklers für eine bestimmte Änderung an einem System festzuhalten.

Leider lassen sich aus simplen Metadaten jedoch häufig mithilfe von Data Mining Techniken mehr Informationen als auf den ersten Blick ersichtlich. Hierbei kann es dazu kommen, dass durch simple Hilfsmittel, welche lediglich die Produktivität und Benutzbarkeit eines Werkzeugs verbessern sollen, private Informationen über den Nutzbar nach außen hin ersichtlich werden. Falls diese Daten nun zusätzlich öffentlich einsehbar sind, können diese von einer nicht kontrollierten Instanz benutzt werden.

Aus diesen Umständen entsteht ein Dilemma, welches aus dem Konflikt zwischen der Notwendigkeit Daten zu erheben und zu veröffentlichen und der unmöglichen Kontrolle über die Verbreitung dieser Daten besteht. Im Folgenden wird dieses Problem in Bezug auf Privatheit und Transparenz am Beispiel der professionellen Open-Source Community und dem Tool Git näher betrachtet.

3 Git und Github

Diese Arbeit erläutert die vorliegende Problematik am Beispiel des Tools *Git* und der Plattform *GitHub*. Hierzu werden im Folgenden die beiden Technologien vorgestellt, wichtige Aspekte erörtert und zusammengefasst.

3.1 Git

Git ist ein heutzutage als Standard angesehenes Werkzeug zum Entwickeln von Projekten im Informationstechnischen Sektor. Beinahe jedes Projekt mit Quellcode besitzt eine Art von Version Control System (VCS), und in den meisten Fällen wird diese Rolle von *Git* eingenommen. Ein VCS ist ein Hilfsmittel, welches es erlaubt den Quellcode zu versionieren, also zu bestimmten Zeitpunkten eine Version des momentanen Standes des Projekts festzuhalten. Ein Projekt, welches von Git verwaltet wird, wird im Fachjargon ein *Repository* genannt.

Durch diese Versionierung, ist es den Entwicklern des Projekts möglich schnell zwischen

verschiedenen Versionen hin und her zu wechseln. Sollte also zum Beispiel ein neues Feature einer Software fehlerhaft sein und nicht funktionieren, kann mithilfe eines Befehls ohne weiteren Aufwand zu der vorherigen stabilen Version des Projektes gewechselt werden. Eine solche Version wird in Git als *Commit* bezeichnet. Ein *Commit* kann nun wiederum auf einen oder mehrere andere *Commits* zeigen, welche die *Parents*, also deren Vorfahren, bezeichnet werden.

Wenn ein neuer *Commit*, also eine neue Version, erstellt wird, zeigt dieser also immer auf die vorherige Version, von der dieser abgeleitet wurde. Durch diese Verbindung zwischen den *Commits* lässt sich folglich die komplette Historie des Projektes ableiten und zu jedem Zeitpunkt des Projektes zurückspringen. Diese Struktur wird in Git die *History* genannt, welche man als gerichteten azyklischen Graphen darstellen kann.

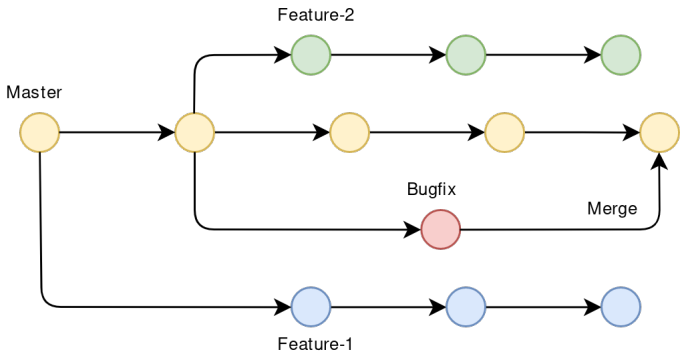

Figure 1: Eine beispielhafte Darstellung einer möglichen *History* in Git.

Wenn man sich nun jedoch einen solchen *Commit* genauer anschaut, fallen einige sehr interessante Details auf. In Listing 1 kann die Struktur einer solchen Datei gesehen werden und es sind mehrere direkte Identifikatoren und Metadaten zu sehen. Zum einen ist der volle Name und die Email Adresse ersichtlich, welche vermutlich zu einer vollständigen Identifikation ausreichen würden. Zudem ist Timestamp mit dem momentanen UTC Offset des Erstellers des *Commits* eingebunden.

```
1    tree      cd7d001b696db430b898b75c633686067e6f0b76
2    parent    c19b969705e5eae0ccca2cde1d8a98be1a1eab4d
3    author    Arne Beer <test@eintest.de> 1513434723 +0100
4    committer Arne Beer <test@eintest.de> 1513434723 +0100
5
6    Chapter 2, acronyms
```

Listing 1: Eine Git *Commit* Datei. Auf unterster Ebene wird ein *Commit* nur durch eine Textdatei in diesem Format dargestellt.

Es ist folglich für jede neue Version zu sehen, wer diese erstellt hat, wann er sie erstellt hat und in welcher Zeitzone sich diese Person zu diesem Zeitpunkt aufgehalten hat. Ebenfalls verbunden mit einem Commit sind alle Änderungen, die der Autor zwischen dieser und der letzten Version vorgenommen hat.

3.2 GitHub

GitHub ist die zur Zeit größte Open-Source Plattform mit über 96 Millionen öffentlichen Repositories. Es ist eine Seite, die zur Verbreitung von Wissen, öffentlicher Software, zum Lernen sowie zum entwickeln kommerzieller Projekte verwendet wird. Neben dem bloßen Bereitstellen einer Plattform, werden zudem Tools angeboten, welche die Kollaboration zwischen Entwicklern vereinfacht. Dadurch wird die Hürde zum Beitragen an fremden Projekten deutlich gesenkt und dementsprechend sogar Kollaboration gefördert.

4 Zusammenfassung

5 Literatur