Kuinka ohjelmoijat etsivät koodia

Jarmo Isotalo

Referaatti
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Helsinki, 11. syyskuuta 2015

Sisältö

1 Johdanto

Projektien koon kasvaessa ja yhä kehittyneempien koodinhakumahdollisuuksien myötä koodin etsimisestä on muodostunut yhä oleellisempi vaihe sekä ohjelmatuotannossa että ylläpitovaiheessa. Sadowski, Stolee ja Elbaum esittelevä artikkelissaan "How Developers Search for Code: A Case Study" [1], kuinka ohjelmoijat etsivät koodia Googlella. He keskittyvät tapaustutkimuksessaan erityisesti tarkastelemaan miksi, miten ja milloin ohjelmoija etsii koodia käyttäen siihen tarkoitettuja työkaluja.

Käyttäen apuna hakutyökalun lokeja sekä juuri ennen koodin hakua tehtyjä kyselyitä he selvittivät vastauksia tutkimuskysymyksiinsä.

2 Tutkimusasetelma

Kirjoittajat kertovat artikkelissaan, että Googlella suurin osa ohjelmakoodista on yhdessä isossa repositoriossa, josta kaikki pääsevät katsomaan ja halutessaan hyödyntämään toistensa koodia. Googlella on myös käytössä sisäinen koodinhakutyökalu (code search tool), jossa hakua voi rajata mm. kielen ja sijainnin perusteella. Hakutulosten näyttämisen lisäksi koodihaku mahdollistaa koodin läpi navigoinnin kansiorakenteen sekä koodiviittausten perusteella.

Tutkimuksdata kerättiin ennen koodin hakua aukeavalla kyselyllä sekä lokianalyysillä. Kyselyä varten he loivat selainlisäosan, joka avaa kyselyn, koodihakua avattaessa. He kuitenkin rajoittivat sekä kyselyiden määrän siten, että ohjelmoijat saavat enintään kymmenen kyselyä vähintään kymmenen minuutin välein. Tällä he pyrkivät vähentämään kyselyn tutkimuksen tuottamaa lisävaivaa tutkittaville, ja siten saamaan laadukkaampia vastauksia. Hakutyökalun lokeista he saavat selville anonymisoidun käyttäjän, tapahtuman kellonajan, hakutermit ja jokaisen käytetyn selaimen välilehden tunnisteen, sekä mitä tuloksia on klikattu. He yhdistivät kyselytulokset sekä lokimerkinnät kellonaikojen perusteella saadakseen tarkemmat vastaukset osiin tutkimuskysymyksistään. Lokianalyysissä lokeista pilkottiin hakusessioita, siten että joka session välissä on vähintään kuuden minuutin tauko. Jokainen sessio koostuu siis koodihaun avaamisesta, hakemisesta, tulosten katselmoinnista ja mahdollisista toistuvista hauista.

Sadowski, Stolee ja Elbaum esittelevät seuraavat tutkimuskysymykset:

1. Miksi ohjelmoijat etsivät koodia
2. Missä kontekstissa haku suoritettiin
3. Millainen tyypillinen hakukysely on
4. Mitä hakusessio tuottaa tulokseksi
5. Miten eri konteksti vaikuttaa hakutuloksiin

3 Johtopäätökset

Tapaustutkimuksesta saatu data sekä niiden yhdiste mahdollistivat tarkemman analyysin siihen, miten koodia etsitään. He myös havaitsivat, että koodia etsitään aktiivisesti, koodihakua käytetään paljon esimerkkien hakuun. Koodia etsitään pääosin jo tutusta koodista. Hakuja myös tarkennetaan useasti iteratiivisesti.

1. **Miksi koodia etsitään?** He kartoittivat syitä koodin etsimiseen lajittelemalla kyselyiden vapaamuotoiset vastaukset omiin ryhmiinsä. Näistä ryhmistä he tunnistivat, että koodia hakemalla pyrittiin vastaamaan seuraaviin kysymyksiin: miten tehdä jotakin, mitä koodi tekee, miksi koodi toimii kuten toimii, missä koodi sijaitsee ja kuka muutti koodia ja milloin.

2. **Missä kontekstissa haku tehtiin?** Koodin hakuun on useita syitä. Tutkimustulosten perusteella 39% hauista tehdään silloin kun työskentelee muutoksen parissa. Myös koodin katselmointivaiheessa sekä ongelman ratkaisussa koodin hakeminen on tyypillistä. Tuloksista heille myös selvisi, että suurin osa hauista tehdään jo osin tuttuun koodiin. Hakuja tehdään myös selvittääkseen miten koodi toimii, miten sitä käytetään (esimerkit yms).

3. **Hakukyselyn ominaisuudet?** He selvittivät lokianalyysillä, millaisia hakukyselyitä tutkimuksen aikana tehtiin. Näistä he havaitsivat, että noin neljännes hauista rajoittaa tuloksia sijainnin perusteella kun taas kielen perusteella hakuja rajattiin vain noin viidessä prosentissa hauista. He myös tunnistivat paljon tilanteita, jossa tehtiin kaksi tai useampi peräkkäinen haku siten, että hakujen välissä ei ollut interaktiota koodihaun kanssa. Tämän he arvelivat johtuvan siitä, että suuri osa peräkkäisistä hauista oli joko kyselyn muokkaamista tai hakualueen rajaamista. He myös havaitsivat, että keskimäärin näiden kahden kyselyn välissä oli vain kahdeksan sekunnin ero; eli tässä ajassa ohjelmoija ehti tarkastelemaan alkuperäiset hakutulokset ja tarkentamaan hakuaan.

4. **Tyypillisen hakusession sisältö?** Lokianalyysistä he havaitsivat, että hakusessio kestää keskimäärin 3 minuuttua ja 30 sekuntia ja sisältää 2 selaimen välilehteä. He havaitsivat myös, että tutkimuksen aikana ohjelmoijat tekivät keskimäärin 12 hakua päivän aikana TODO: patterns.

5. **Kontekstin vaikutus hakumenetelmiin (pattern)?** Kyselyistä ja lokianalyysistä saamillaan tuloksia yhdistämälle he pystyivät tunnistamaan tarkempia hakumenetelmiä. Mikäli ohjelmoija halusi selvittää koodin ominaisuuksia he harvoin klikkasivat yhtään tiedostoa, tämän

he uskoivat johtuvan siitä, että vastaus näkyi hakutulosten esikatselussa, tai koska haku tuotti vain yhden tuloksen ja näytti koko tiedoston siinä. He myös vahvistivat oletuksensa, että ohjelmoija joka ei tunne hakemaansa koodia kunnolla navigoi ja hakee siihen liittyen enemmän kuin koodin tunteva henkilö.

co:

4 Yhteenveto

Tulosten pohjalta heillä on koodinhakutyökalujen luojille muutamia ehdoituksia:

- Tarjoa tuloksien esikatselu.
- Tarjoa mahdollisuus filtteröidä tiedoston sijainnin, kielen ja/tai tekijöiden perusteella.
- Tarjoa tulokset huomioiden monipuolisempi konteksti, mahdollisesti jopa seuraamalla käyttäjän toimia muualla, mm. käydyt keskustelut.
- Harkitse työkalun integraatiota kehitysympäristöön.

Sadowski, Stolee ja Elbaum muistuttavat vielä lopuksi, että Googlen uniikki toimintatapa saattaa vaikuttaa tulosten soveltuvuuteen ja toistettavuuteen.

Lähteet

[1] Sadowski, Caitlin, Stolee, Kathryn T. ja Elbaum, Sebastian: *How Developers Search for Code: A Case Study.* Teoksessa *Joint Meeting of the European Software Engineering Conference and the Symposium on the Foundations of Software Engineering (ESEC/FSE)*, 1600 Amphitheatre Parkway, 2015.