

Cuprins

1 Searcher	3
1.1 Clauze	3
1.1.1 Comparare prin semne	3
1.1.2 Comparare prin limbaj natural	4
1.1.3 Restricție de timp	4
1.1.4 Apartenență	5
1.1.5 Restricție asupra fișierelor	5
1.1.6 Restricție asupra autorilor	6
1.2 Interogări simple	7
1.2.1 Fișiere	7
1.2.2 Autori	12
1.3 Interogări despre evoluție	15
1.3.1 Fișiere	15
1.3.2 Autori	17
1.4 Interogări comune	21
1.5 Combinarea și negarea interogărilor	22

Capitolul 1

Searcher

Searcher este un limbaj dedicat creat pentru a ajuta la analiza sistemelor. Acest limbaj permite folosirea unor interogări pentru fișiere și pentru autori dintr-un sistem software.

1.1 Clauze

Searcher are multiple clauze care pot fi folosite la crearea unor interogări pentru a oferi o filtrare cât mai amplă și specifică.

1.1.1 Comparare prin semne

Clauza de comparare prin semne permite interogarea valorii câmpurilor și compararea acestora cu un parametru. Semnele posibile pentru această clauză sunt: >, <, >=, <=, = sau folosind cuvântul cheie "in" urmat de un interval: [],], [, (). Numerele care pot fi folosite în aceste comparații trebuie să fie numere naturale. În interval paranteza dreaptă înseamnă că numărul este inclus în interval, iar paranteza rotundă reprezintă excluderea numărului din interval. Intervalul se scrie sub următoarea formă: in [valoareMinimă, valoareMaximă], iar *valoareMinimă* trebuie să fie mai mic sau egal cu *valoareMaximă* pentru a produce un rezultat. Clauza de comparare prin semne se scrie întotdeauna după câmpul care se dorește a fi comparat.

Exemple de clauze de comparare prin semne:

```
file: size >= 500  
file: size in (400, 2000]
```

1.1.2 Comparare prin limbaj natural

Această clauză oferă o abordare mult mai naturală la compararea valorilor câmpurilor cu un parametru specific. Cuvintele cheie care pot fi folosite în această clauză sunt: `moreThan`, `lessThan`, `atLeast`, `atMost` și `exactly` sau folosind cuvântul cheie “between” urmat de un interval: `[]`, `]`, `(]`, `()`. Parantezele drepte reprezintă incluziunea numărului în interval, iar cele rotunde reprezintă excluderea numărului din interval. Intervalul se scrie sub următoarea formă: `between [valoareMinimă, valoareMaximă)`, iar *valoareMinimă* trebuie să fie mai mic sau egal cu *valoareMaximă* pentru a produce un rezultat. Cuvintele cheie folosite în această clauză de comparare au următoarele semnificații:

`moreThan` -> mai mult de
`lessThan` -> mai puțin de
`atMost` -> cel mult
`atLeast` -> cel puțin
`exactly` -> exact

O clauză de comparare prin limbaj natural se scrie întotdeauna și în mod natural înaintea câmpului care se dorește a fi comparat.

Exemple de clauze prin limbaj natural:

```
file : has moreThan 50 commits  
file : has between [20, 30) developers
```

1.1.3 Restricție de timp

Clauza de restricție de timp permite restricționarea intervalului de timp în care câmpurile sunt acceptate. Clauza se scrie întotdeauna după câmpul care se dorește a fi căutat și poate fi combinată cu orice altă clauză. Cuvintele cheie care pot fi folosite în această clauză sunt: `since`, `until`, `before` și `after` sau folosind cuvântul cheie “between” urmat de doua date separate prin virgulă sub această formă: `between 2011, 2015-09-20`. Pentru a putea oferi un rezultat prima dată trebuie să fie mai veche iar cea de-a doua trebuie să fie mai recentă decât prima. Cuvintele cheie folosite în această clauză de comparare au următoarele semnificații:

`after` -> după dată

before -> înainte de dată

since -> de la data de

until -> până la data de

Data folosită în aceste clauze pentru a restricționa intervalul de timp are următoarele forme: AAAA, AAAA-LL, AAAA-L, AAAA-LL-ZZ, AAAA-LL-Z, AAAA-L-ZZ, AAAA-L-Z

(unde AAAA reprezintă anul, LL sau L reprezintă luna cu două sau cu o cifră, ZZ sau Z reprezintă ziua cu două sau cu o cifră)

Exemple de clauze de restricție de timp:

```
file: has changes after 2018
```

```
file: has moreThan 10 commits since 2017-07-28
```

```
file: has developers >= 19 between 2011, 2015-09-30
```

1.1.4 Apartenență

Această clauză permite căutarea câmpului de interes într-un anumit set de fișiere sau de autori. Setul respectiv de fișiere sau de autori poate să fie o listă sub această formă: ["unNume", "altNume"], poate să fie o altă interogare sub forma: {altăInterogare} sau poate să fie un nume de interogare salvată sub următoarea formă: {{numeInterogareSalvată}}. În listă, între paranteze, în ghilimele separate prin virgulă se pot adăuga oricâte nume, acest exemplu conținând doar două nume. Interogările scrise pot fi salvate cu un anumit nume iar după aceea se poate folosi direct numele respectiv fără a mai trebui scrisă aceeași interogare încă o dată. Sintaxa acestei clauze este cuvântul cheie "in" urmat de una din cele trei forme ale setului (listă, altă interogare sau un nume de interogare), iar această clauză urmează întotdeauna după câmpul care se dorește a fi căutat.

Exemple de clauze de apartenență:

```
file: owner in ["John", "Mike"]
```

```
file: owner in {author: name contains "son" }
```

```
file: owner in {{RetiredAuthorsWithBigKnowledge}}
```

1.1.5 Restricție asupra fișierelor

Clauza de restricție asupra fișierelor se poate folosi doar la interogări legate de fișiere și anume doar la acele interogări care conțin

cuvintele cheie *changes*, *commits* sau *churn*. Această clauză permite restricționarea autorilor, care sunt responsabili pentru câmpurile după care se caută. Clauza se scrie întotdeauna după câmpul care se dorește a fi căutat și poate fi combinată cu orice altă clauză. Această clauză poate fi scrisă în mai multe moduri, toate conținând cuvântul cheie "madeBy". Modurile în care se pot scrie această clauză sunt: madeBy "unNume", madeBy any, madeBy any clauză de apartenență, madeBy everyOne, madeBy everyOne clauză de apartenență, madeBy clauză de comparare prin limbaj natural, madeBy clauză de comparare prin limbaj natural clauză de apartenență. Cuvântul cheie *any* reprezintă orice autor iar *everyOne* reprezintă toți autorii. În cazul folosirii clauzei de apartenență, dacă se dorește să se scrie o altă interogare sau se dorește să se folosească o interogare salvată, interogarea trebuie să fie neapărat o interogare de autori.

Exemple de clauze de restricție asupra fișierelor:

```
file: has changes madeBy "John"
file: has changes madeBy everyOne in ["John", "Mike"]
file: has changes madeBy any in {{
ActiveAuthorsWithBigKnowledge}}
file: has changes since 2015 madeBy moreThan 2 in {author:
name startsWith "Mi"}
```

1.1.6 Restricție asupra autorilor

Această clauză poate fi folosită doar la interogări legate de autori și anume doar la acele interogări care conțin cuvintele cheie *changes*, *commits* sau *churn*. Această clauză permite restricționarea fișierelor pe care autorii au lucrat și le-au modificat. Clauza se scrie întotdeauna după câmpul care se dorește a fi căutat și poate fi combinată cu orice altă clauză. Această clauză poate fi scrisă în mai multe moduri, toate conținând cuvântul cheie "on". Modurile în care se pot scrie această clauză sunt: on "unNume", on any, on any clauză de apartenență, on every, on every clauză de apartenență, on clauză de comparare prin limbaj natural, on clauză de comparare prin limbaj natural clauză de apartenență. Cuvântul cheie *any* reprezintă orice fișier, iar *every* reprezintă toate fișierele. În cazul folosirii clauzei de apartenență, dacă se dorește să se scrie o altă interogare sau se dorește să se folosească o interogare salvată, interogarea trebuie să fie neapărat o interogare de

fișiere.

Exemple de clauze de restricție asupra autorilor:

```
author: has moreThan 50 changes between 2013, 2014 on "
KafkaClient.java"
author: has churn > 10 on any in ["KafkaAdminClient.java",
"KafkaConsumer.java"]
author: has changes on every in { file: name endsWith ".java"
}
author: has commits on moreThan 5 in { file: name startsWith
"Kafka" }
```

1.2 Interogări simple

Moduri în care se pot scrie interogări simple pentru fișiere și pentru autori.

1.2.1 Fișiere

Interogările simple pentru fișiere, din limbaj, sunt cele care oferă informații despre numele sau calea unui fișier, dimensiunea acestora, data la care au fost scrise, proprietarii fișierelor, codul din fișiere, apartenența fișierelor, relațiile dintre fișiere și dacă folosesc bibliotecile *library* sau *language*. Aceste interogări sunt simple de scris și oferă foarte multe informații esențiale despre sistemul analizat mai ales dacă se scrie o interogare în care mai multe expresii simple sunt combinate, acest aspect va fi prezentat în secțiunea 1.5, pentru a obține rezultate specifice și concise.

Interogările de fișiere trebuie neaparat să înceapă cu cuvântul cheie *file* urmat de două puncte deoarece astfel se marchează tipul de interogare care urmează să fie scris, iar prin cuvântul *file* se specifică folosirea unei interogări care are legătură cu fișierele din sistem.

Fișierele sistemului pot fi filtrate în mai multe feluri scriind interogări care caută după numele sau după calea fișierelor. Se pot scrie interogări care să caute fișiere care conțin un anumit șir de caractere în numele lor, care au șirul în numele lor, care încep sau se termină cu acest șir de caractere sau care au exact șirul acela de caractere ca și nume. De asemenea se pot scrie și expresii regulate, iar în funcție de expresia regulată scrisă se vor căuta fișierele care corespund acestei expresii.

O interogare de nume sau de cale se scrie în felul următor: se începe cu unul din cele două cuvinte cheie *name* sau *path* care marchează fie legătura cu numele fișierelor fie cu calea fișierelor. După aceea se poate alege unul din următoarele cuvinte *has*, *contains*, *is*, *matches*, *startsWith* sau *endsWith*. Ultima parte a interogării poate fi ori *un șir de caractere* ori *o clauză de apartenență*. Clauza de apartenență în această interogare poate fi doar de tipul celei cu listă, nu și celelalte două variante, iar între cuvintele menționate mai sus și clauza de apartenență trebuie adăugat alt cuvânt cheie și anume *any*. *Has* va căuta fișierele care au șirul de caractere scris, după cuvântul cheie, în numele sau calea lor, depinde ce s-a ales să fie căutat. *Contains* va căuta fișierele care conțin șirul de caractere în numele sau calea lor. *Is* va căuta fișierele care au acest șir de caractere exact ca și nume sau ca și cale. *Matches* va căuta fișierele după o expresie regulată. *StartsWith* și *endsWith* vor căuta numele sau calea fișierelor care încep sau care se termină cu șirul de caractere scris. Aceleași rezultate se vor produce și dacă se folosește clauza de apartenență cu listă. În loc de căutarea unui singur șir de caractere se vor căuta fișierele care se potrivesc cu oricare din șirurile de caractere din lista scrisă.

Există și posibilitatea de a scrie această interogare și mai simplu fără cuvintele *has*, *contains*, *is*, *matches*, *startsWith* sau *endsWith*. Interogarea va conține doar unul din cuvintele cheie *name* sau *path* pentru a alege dacă se dorește căutarea după nume sau după cale și o clauză de apartenență de tipul celei cu listă. Această interogare va căuta fișierele care au oricare din șirurile de caractere din listă ca nume sau ca și cale a lor.

Exemple:

```
file: name has "Main"  
file: path contains "Impl"  
file: name startsWith "Kaf"  
file: path endsWith any in [".java", ".scala"]  
file: name in ["Sender.java", "KafkaProducer.java"]
```

O altă categorie importantă de filtrare a fișierelor o reprezintă dimensiunea acestora. De obicei fișierele care au dimensiuni mai mari sunt mai predispuse să fie fișiere probleme, deoarece ele conțin multe informații și dacă în viitor ceva trebuie schimbat sau o funcționalitate nouă trebuie adăugată, o mică modificare în aceste fișiere mari poate provoca multe erori în componenta respectivă sau și în tot sistemul, în

cel mai rău caz. De aceea, filtrarea fișierelor după dimensiune este una esențială pentru a asigura mentenanța sistemului. Interogările de dimensiune se scriu astfel: cuvântul cheie *size* urmat de o clauză de comparare prin semne.

Exemple:

```
file: size > 500
file: size in (300, 1000]
```

Se pot scrie interogări care caută după data creării unui fișier sau după data ultimei modificări. Astfel se pot filtra fișierele după data la care au fost create și se mai pot filtra fișierele pentru a vedea când a fost făcută ultima modificare asupra lor. Cu aceste interogări se pot determina care sunt fișierele care nu s-au modificat de mult și care sunt posibile să fie "moarte" sau pe de altă parte instabile, deoarece o modificare minoră ar produce multe erori neașteptate și de aceea nu s-au mai produs schimbări în acele fișiere, sau se pot determina fișierele cele mai vechi și cele mai recente din sistem. Aceste interogări au următoarea structură: unul din cuvintele cheie *created*, pentru a căuta după creare, sau *lastChanged*, pentru a căuta după ultima modificare, urmat de o clauză de restricție de timp.

Exemple:

```
file: created between 2015-03-10, 2015-03-22
file: lastChanged before 2017
```

Pentru a vedea care persoane ce fișiere dețin se pot scrie interogări astfel: cuvântul cheie *owner* urmat de un șir de caractere sau de o clauză de apartenență. În aceste interogări se pot folosi toate tipurile de clauză de apartenență. Cu această interogare se pot găsi toate fișierele unui proprietar iar astfel se poate vedea câte fișiere deține fiecare persoană din sistem. Interogările care se pot scrie în clauza de apartenență din aceste interogări trebuie să fie de tipul *author*: (trebuie să fie interogări de autori), iar numele de interogări folosite trebuie să corespundă, la fel, unor interogări de autori.

Exemple:

```
file: owner "John"
file: owner in ["Josh", "Hope"]
file: owner in {author: name endsWith "son"}
file: owner in {{PassiveAuthorsWithBigKnowledge}}
```

Se poate căuta și după codul din fișiere. Astfel se pot căuta fișierele care conțin o bucată de cod sau doar un cuvânt anume, sau se poate căuta în codul fișierelor cu ajutorul unei expresii regulate. Pentru a căuta în conținutul fișierelor o bucată de cod sau un cuvânt se va folosi cuvântul cheie *contains*, iar pentru folosirea unei expresii regulate se va folosi cuvântul cheie *matches*. O astfel de interogare se scrie în felul următor: cuvântul cheie *code* urmat de unul din cuvintele *contains* sau *matches* iar după aceea urmează un șir de caractere sau o expresie regulată depinde ce cuvânt s-a ales înainte. La sfârșitul acestei interogări se poate adăuga și o *clauză de comparare prin semne* sau o *clauză de comparare prin limbaj natural* pentru a căuta dacă șirul de caractere apare de un anumit număr de ori în conținutul fișierelor. Dacă clauzele lipsesc atunci se vor afișa toate fișierele care au șirul de caractere în conținut cel puțin o dată.

Exemple:

```
file: code contains "main"  
file: code contains "if" moreThan 10  
file: code contains "static" < 5
```

Alt tip de interogări care pot fi scrise sunt interogările care caută după fișiere care aparțin cuiva sau care au un tag special, și anume un component tag. Un component tag se scrie cu *hashtag* (#) urmat de cuvântul *component*. Un component tag poate fi compus din mai multe sub tag-uri pentru a verifica apartenența la un component tag mult mai specific. Sub tag-urile se adaugă la component tag prin punct, de exemplu: *#component.subTag*. Se pot scrie oricâte alte sub tag-uri, iar concatenarea lor se face prin punct. De asemenea după component tag se poate folosi steluță pentru a marca astfel toate sub tag-urile component tag-ului, exemplu: *#component.**.

Există trei feluri de a scrie o astfel de interogare. Prima variantă se referă la căutarea unei apartenențe între fișiere și o persoană anume. Această interogare se scrie astfel: cuvântul cheie *belongsTo* urmat de un șir de caractere. A doua variantă reprezintă căutarea apartenențelor între fișiere și component tag, iar acest lucru se scrie astfel: cuvântul cheie *belongsTo* urmat de *#component* (component tag), la component tag se pot concatena oricâte alte sub tag-uri sau se poate folosi steluță pentru a marca toate sub tag-urile. Ultima variantă corespunde căutarea unei apartenențe între fișiere și o listă de component tag-uri. Interogarea se scrie astfel: cuvântul cheie *belongsTo* urmat de unul din cuvintele *any* sau *every* sau de o *clauză*

de comparare prin limbaj, urmat de cuvântul *in*, iar ultima parte reprezintă lista de component tag-uri. În această listă se pot scrie oricâte component tag-uri separate prin virgulă. Folosirea cuvântului cheie *any* reprezintă prezența unei apartenențe cu oricare component tag din listă, iar *every* reprezintă prezența unei apartenențe cu toate component tag-urile din listă.

Exemple:

```
file: belongsTo "John"
file: belongsTo #component.subTag
file: belongsTo any in [#component.subTag, #component.
altSubTag.*]
file: belongsTo every in [#component.subTag, #component.
altSubTag]
file: belongsTo exactly 1 in [#component.subTag, #component
.altSubTag]
```

Pe lângă căutarea apartenenței fișierelor se poate căuta și după legătura lor cu anumite persoane sau cu alte fișiere. Acest lucru permite determinarea cuplajului între fișiere dar și legăturile fișierelor cu anumite persoane. Aceste interogări se scriu în felul următor: cuvântul cheie *isRelatedTo* sau *hasRelationTo*, ambele oferă aceleași rezultate se poate folosi oricare cuvânt, urmat de *un șir de caractere* sau de *o clauză de apartenență*. În această interogare se poate folosi oricare variantă a clauzei de apartenență. Interogările care se pot scrie în clauza de apartenență din aceste interogări trebuie să fie de tipul *file*: (trebuie să fie interogări de fișiere), iar numele de interogare folosite trebuie să corespundă, la fel, unor interogări de fișiere.

Exemple:

```
file: isRelatedTo "John"
file: hasRelationTo in ["Mike", "Josh"]
file: isRelatedTo in {file: name contains "Impl"}
file: hasRelationTo in {{FilesWithSizeBiggerThan700}}
```

Se pot scrie și interogări care caută fișierele care folosesc o bibliotecă și anume biblioteca *library* sau *language*. Aceste interogări se pot scrie în două feluri. Prima variantă reprezintă căutarea fișierelor după folosirea unei biblioteci și are următoarea structură: cuvântul cheie *uses* urmat de *hashtag* și unul din bibliotecile după care se caută (*library* sau *language*). Aceste biblioteci se scriu la fel ca și un component tag. Se pot concatena sub biblioteci la cele principale prin

folosirea unui punct sau se poate folosi steluță pentru selectarea tuturor sub bibliotecilor. A doua variantă corespunde căutării fișierelor folosind o listă de biblioteci. Interogarea se scrie astfel: cuvântul cheie *uses* urmat de unul din cuvintele *any* sau *every* sau de o clauză de comparare prin limbaj, urmat de cuvântul *in*, iar ultima parte reprezintă lista de biblioteci. În această listă se pot scrie oricâte biblioteci separate prin virgulă. Folosirea cuvântului cheie *any* reprezintă folosirea oricărei biblioteci din listă, iar *every* reprezintă folosirea tuturor bibliotecilor din listă.

Exemple:

```
file: uses #library.subBiblioteca
file: uses any in [#library.subBiblioteca, #language.*]
file: uses every in [#library.subBiblioteca, #library.
altaSubBiblioteca]
file: uses exactly 1 in [#language.subBiblioteca, #library.
altaSubBiblioteca]
```

1.2.2 Autori

Interogările simple pentru autori, din limbaj, sunt cele care oferă informații despre numele unui autor, activitatea acestora, PKS-ul (*Project Knowledge Score*) pe care îl au, când au început să lucreze pe un fișier și când au commit-uit ultima dată. Interogările acestea, care deși se scriu într-o manieră ușoară, oferă informații importante despre autori și despre implicarea acestora în sistemul analizat. De asemenea aceste interogări pot fi combinate, aspect care va fi prezentat în secțiunea 1.5, pentru a obține informații specifice despre autorii sistemului.

Interogările de autori trebuie neapărat să înceapă cu cuvântul cheie *author* urmat de două puncte deoarece astfel se marchează tipul de interogare care urmează să fie scris, iar prin cuvântul *author* se specifică folosirea unei interogări care are legătură cu autorii sistemului.

Autorii sistemului pot fi filtrați în mai multe feluri scriind interogări care caută după numele acestora. Se pot scrie interogări care să caute autori care conțin un anumit șir de caractere în numele lor, care au șirul în numele lor, care încep sau se termină cu acest șir de caractere sau care au exact șirul acela de caractere ca și nume. De asemenea se

pot scrie și expresii regulate, iar în funcție de expresia regulată scrisă se vor căuta autorii care corespund acestei expresii.

O interogare de nume pentru autori are următoarea formă: cuvântul cheie *name*, iar după aceea se poate alege unul din următoarele cuvinte *has*, *contains*, *is*, *matches*, *startsWith* sau *endsWith*. Ultima parte a interogării poate fi ori un șir de caractere ori o clauză de apartenență. Clauza de apartenență în această interogare poate fi doar de tipul celei cu listă, nu și celelalte două variante. *Has* va căuta autorii care au șirul de caractere scris, după cuvântul cheie, în numele lor. *Contains* va căuta autorii care conțin șirul de caractere. *Is* va căuta autorii care au acest șir de caractere exact ca și nume. *Matches* va căuta autorii după o expresie regulată. *StartsWith* și *endsWith* vor căuta numele autorilor care încep sau care se termină cu șirul de caractere scris. Aceleași rezultate se vor produce și dacă se folosește clauza de apartenență cu listă. În loc de căutarea unui singur șir de caractere se vor căuta autorii care se potrivesc cu oricare din șirurile de caractere din lista scrisă.

Exemple:

```
author: name contains "son"  
author: name startsWith "Jo"  
author: name in ["John", "Mike"]
```

Autorii pot fi căutați și după activitatea lor. Astfel se pot vedea care autori din sistem sunt încă activi, care sunt pasivi sau care s-au retras de pe proiect. Interogările de activitate se scriu astfel: cuvântul cheie *is* urmat de unul din cuvintele *ACTIVE*, autori activi din sistem, *PASSIVE*, cei pasivi, sau *RETIRED*, autori retrași de pe proiect.

Exemple:

```
author: is ACTIVE  
author: is PASSIVE  
author: is RETIRED
```

De asemenea autorii pot fi filtrați după PKS-ul lor. PKS (*Project Knowledge Score*) reprezintă cât la sută cunoștințe despre proiect are fiecare autor. Prin aceste interogări se pot căuta și vizualiza autorii care au mari cunoștințe despre proiect, ei reprezentând o parte importantă din echipa care lucrează pe proiect având multe informații despre acesta. Interogările acestea au următoarea formă: cuvântul cheie *hasPKS* urmat de o clauză de comparare prin semne, iar la sfârșit după scrierea numărului cu care se face compararea trebuie scris semnul de procente (%). Aici pentru aceste interogări în clauza de comparare prin semne se pot folosi numere reale pozitive, nu trebuie folosite doar numere naturale, iar partea cu folosirea unui interval de numere în acest de tip de interogări nu este valabilă.

Exemple:

```
author: hasPKS > 5%  
author: hasPKS <= 57.89%  
author: hasPKS = 23.5%
```

Se pot căuta autori și după data când au început să lucreze sau după data ultimului commit făcut. Prin aceste interogări se pot găsi autorii sistemului care sunt de mai mult timp pe proiect sau autori care sunt noi veniți și au început de puțin timp să lucreze pe proiect. Interogările se scriu astfel: cuvântul cheie *started* sau *lastCommitted*, depinde după ce criteriu se dorește filtrarea, urmat de o clauză de restricție de timp. Cuvântul cheie *started* reprezintă căutarea după data începerii lucrului, iar *lastCommitted* reprezintă căutarea după

data ultimului commit făcut de către autori.

Exemple:

author: `started between 2015-03-01, 2015-04-01`

author: `lastCommitted before 2017`

1.3 Interogări despre evoluție

Interogările care sunt legate de evoluția fișierelor și a autorilor din sistemul analizat.

1.3.1 Fișiere

Pe lângă interogările pentru fișiere prezentate în secțiunea 1.2.1, limbajul oferă și posibilitatea scrierii unor interogări care oferă informații despre schimbările fișierelor, creșterea lor, numărul de dezvoltatori ai fiecărui fișier, numărul de commit-uri sau de task-uri pe care un fișier le are și schimbările, commit-urile sau churn-ul făcute de cineva pe fișiere.

Interogările care oferă informații despre schimbările fișierelor sunt esențiale pentru a face o analiză completă asupra sistemelor. Schimbările arată cât de stabile sau instabile sunt fișierele respective. Arată care fișiere au avut probleme și au trebuit să fie modificate de-a lungul timpului.

Aceste interogări se pot scrie în mai multe feluri combinând mai multe clauze ale limbajului între ele. Un prim fel de a scrie aceste interogări este următorul: cuvântul cheie *has* urmat de o posibilă clauză de comparare prin limbaj natural, iar după aceea cuvântul *changes* urmând o eventuală clauză de restricție de timp. Dacă clauza de comparare prin limbaj natural lipsește se vor afișa toate fișierele care au cel puțin o modificare, iar dacă clauza de restricție de timp lipsește nu se va impune nicio restricție de dată returnându-se toate fișierele care îndeplinesc interogarea din orice dată.

A doua variantă de a scrie interogarea este foarte asemănătoare cu prima. Se vor scrie cuvintele cheie *has* și *changes* unul după altul iar ele pot fi urmate de o clauză de comparare prin semne și de o clauză de restricție de timp, scrise în această ordine. Ca și în primul caz, ambele clauze sunt opționale, iar dacă acestea lipsesc nu se va impune niciun

fel restricție și se vor returna fișierele cu cel puțin o modificare din orice dată.

În ambele cazuri se poate scrie doar una din cele două clauze, nu trebuie să fie împreună prezente sau să lipsească deodată din interogare.

A treia variantă este mai scurtă de scris și are următoarea formă: cuvântul cheie *changed* urmat de o posibilă clauză de comparare prin limbaj natural și de o posibilă clauză de restricție de timp, scrise în ordinea prezentată. Și în acest caz clauzele sunt optionale și prezența sau absența lor nu trebuie să fie simultană, se poate scrie doar o singură clauză.

Toate cele trei variante urmăresc găsirea fișierelor cu modificări din sistem, diferența majoră dintre ele însă va fi prezentată în cele ce urmează de-a lungul acestei secțiuni.

Exemple:

```
file: has moreThan 10 changes since 2015-03-16
file: has changes after 2017
file: has moreThan 15 changes
file: has changes >= 22 before 2018-11-21
file: has changes < 50
file: has changes until 2018-12
file: changed moreThan 10 since 2018
file: changed exactly 12
file: changed before 2017-05-12
```

Pentru a putea căuta după numărul de commit-uri pentru fiecare fișier, numărul de dezvoltatori care au lucrat pe fișiere, numărul de task-uri de pe fișiere și cu cât au crescut fișierele, se vor folosi primele două variante menționate mai sus. În toate aceste cazuri ordinea cuvintelor și a clauzelor rămâne la fel ca și în cazul interogărilor pentru schimbările fișierelor. La fel, clauzele sunt optionale. Diferența pentru a schimba căutarea fișierelor de la modificări la una din cele amintite mai sus este înlocuirea cuvântului cheie *changes*. Pentru a căuta după commit-uri se va folosi cuvântul *commits*, pentru a căuta după dezvoltatori se va folosi cuvântul *developers*, pentru a căuta după task-uri se va folosi cuvântul *tasks*, iar pentru a căuta după creșterea fișierelor se va folosi cuvântul *growth*.

Pe lângă toate aceste interogări se mai poate căuta și după churn-ul fișierelor. Churn-ul poate fi definit ca și numărul de linii adăugate, modificate sau șterse dintr-un fișier de la o versiune a lui la alta. Aceste

interogări se scriu ca și cele care caută schimbări, dar doar din varianta a doua. Prima varianta cu clauza de comparare prin limbaj nu este valabilă, doar cea cu clauza de comparare prin semne. De asemenea pentru a putea căuta după churn-ul fișierelor se va înlocui cuvântul *changes* cu *churn*, iar clauza de comparare prin semne care la schimbări este opțională, în cazul acesta la churn va fi obligatorie.

Se mai poate căuta și creșterea fișierelor într-o manieră mai simplă, ca și în cazul interogărilor de schimbări din varianta a treia, singura diferență va fi înlocuirea cuvântului *changed* cu *grew*.

Exemple:

```
file: has moreThan 10 commits since 2015-03-16
file: has tasks after 2017
file: has moreThan 15 developers
file: has growth >= 200 before 2018-11-21
file: has commits < 50
file: grew moreThan 100 since 2018
```

Interogările care conțin cuvintele cheie *changes*, *commits* sau *churn* sunt mai speciale și pot avea la sfârșitul lor încă o clauză în plus, și anume o clauză de restricție asupra fișierelor. Clauza se referă la modificările, commit-urile sau churn-ul făcute de cineva pe fișierele sistemului. Această clauză este opțională, iar la absența ei nu va exista nicio restricție referitoare la cine a lucrat pe acele fișiere. Structura acestei clauze poate fi văzută în secțiunea 1.1.5.

Exemple:

```
file: has moreThan 10 changes since 2015-03 madeBy "John"
file: has commits after 2017 madeBy any in {author: name
startsWith "Mi"}
file: has churn > 200 madeBy any in ["Mike", "Kristen"]
file: has commits >= 5 before 2018-11-21 madeBy everyOne in
{{ActiveAuthorsWithBigKnowledge}}
file: has churn < 500 madeBy any
file: has changes until 2018 madeBy moreThan 2 in ["John",
"Mike", "Kristen"]
```

1.3.2 Autori

În secțiunea 1.2.2 au fost prezentate interogările simple pentru autori. Pe lângă acestea, Searcher oferă posibilitatea scrierii și a altor

interogări care sunt legate de fișierele, componentele sau task-urile pe care autorii au lucrat sau le-au modificat, care caută după smartCommit-urile autorilor, care filtrează autorii după perioada în care au lucrat și care caută autorii care au făcut schimbări, au commit-uit sau au churn pe anumite fișiere.

Primele interogări care vor fi prezentate sunt cele care au legătură cu autorii care au lucrat pe anumite fișiere. Cu aceste interogări se pot filtra autorii după fișierele pe care au lucrat. Prin aceste filtrări se poate vedea care autori au lucrat pe un fișier anume sau pe mai multe fișiere din sistemul care este analizat.

Interogările se scriu în felul următor: cuvântul cheie *workedOn* care poate fi urmat de un șir de caractere pentru a indica numele unui singur fișier sau de o clauză de comparare prin limbaj natural. După șirul de caractere, dacă se alege această variantă de a scrie interogarea, poate urma o clauză de restricție de timp, care este opțională. Dacă se alege varianta cu clauza de comparare, după aceasta trebuie scris cuvântul *files*. Apoi, la sfârșitul acestei interogări, după cuvântul *files* poate urma o clauză de apartenență, iar după această clauză poate urma și o clauză de restricție de timp. Ambele clauze de la sfârșitul acestei variante de interogare sunt opționale, iar ele nu trebuie să fie prezente deodată sau să lipsească ambele, se poate scrie doar una dintre ele. Dacă se scriu ambele atunci trebuie respectată ordinea lor, care este menționată mai sus. Interogările care se pot scrie în clauza de apartenență din aceste interogări trebuie să fie de tipul *file*: (trebuie să fie interogări de fișiere), iar numele de interogare folosite trebuie să corespundă, la fel, unor interogări de fișiere.

Pentru a putea filtra autorii după fișierele pe care le-au modificat și nu după fișierele pe care au lucrat, se va face înlocuirea cuvântului cheie *workedOn* cu *modified*. Astfel se face distincția dintre cele două tipuri de filtrare. Interogările se vor scrie la fel ca și mai sus, doar cuvântul *workedOn* trebuie schimbat cu *modified*.

Exemple:

```
author: modified "Sender.java"  
author: workedOn "Sender.java" after 2017-10-18  
author: modified moreThan 10 files since 2018  
author: workedOn lessThan 20 files in {file: name endsWith  
".java"}
```

Pe lângă scrierea unor interogări care caută după autorii care au lucrat pe fișiere, Searcher oferă și posibilitatea de a căuta după autorii care au lucrat pe componentele sistemului. Prin aceste interogări se poate crea o privire de ansamblu asupra autorilor din sistem. Cu ajutorul lor se poate observa care autori au lucrat pe mai multe componente, oferind astfel informații despre implicarea autorilor în mai multe părți din sistem. Interogările se scriu în felul următor: cuvântul *workedOn* urmat de o clauză de comparare prin limbaj natural și de cuvântul cheie *components*. La sfârșitul acestor interogări se poate scrie și o clauză de restricție de timp, care este opțională, la lipsa ei nu se va impune nicio restricție de dată.

La fel ca și în cazul interogărilor în care se caută după fișierele pe care autorii au lucrat sau le-au modificat, și aici la componente se poate înlocui cuvântul cheie *workedOn* cu *modified*, pentru a obține interogări care caută autorii care au modificat componente din sistem.

Exemple:

author: *workedOn* *moreThan* 3 *components*

author: *modified* *exactly* 2 *components* *since* 2018

De asemenea, se pot scrie interogări care caută după autorii care au lucrat pe task-uri. Ele se scriu astfel: cuvântul *workedOn* urmat de o clauză de comparare prin limbaj natural și de cuvântul cheie *tasks*. La sfârșitul acestor interogări se poate scrie și o clauză de restricție de timp, care este opțională, la lipsa ei nu se va impune nicio restricție de dată.

Exemple:

author: *workedOn* *moreThan* 4 *tasks*

author: *workedOn* *lessThan* 7 *tasks* *since* 2018-03-01

Searcher mai oferă și posibilitatea de a căuta autorii care au smartCommit-uri. Aceste interogări se scriu în felul următor: cuvântul *has* urmat de o clauză de comparare prin limbaj natural și de cuvântul cheie *smartCommits*. La sfârșitul interogărilor se poate scrie și o clauză de restricție de timp, care este opțională, la lipsa ei nu se va impune nicio restricție de dată.

Exemple:

author: *has* *moreThan* 4 *smartCommits*

author: *has* *exactly* 3 *smartCommits* *after* 2018-03-01

Se mai pot căuta și autorii care au lucrat într-o perioadă anume. Astfel se pot vedea autorii care au lucrat în diferite faze și momente ale sistemului. Ele au următoarea formă: cuvântul *worked* urmat de o clauză de restricție de timp. În acest caz, clauza nu mai este opțională, ea devine obligatorie. Pe lângă cuvântul *worked* se mai poate folosi și cuvântul *wasActive* pentru a căuta autorii care au fost activi într-o perioadă anume.

Exemple:

author: *worked between* 2017-01-01, 2017-03-01

author: *wasActive after* 2018-09-11

În majoritatea cazurilor schimbările produc multe probleme în orice sistem software, mai ales în cele mari și complexe în care lucrează și multe persoane, dar și în cele de dimensiuni mai mici care au și un număr mai redus de dezvoltatori. Pentru a oferi posibilitatea de a efectua o analiză cât mai completă, Searcher permite scrierea unor interogări care caută autorii care au făcut modificări în sistem.

Aceste interogări se scriu astfel: cuvântul *has* urmat de o posibilă clauză de comparare prin limbaj natural, ea fiind opțională, după care se scrie cuvântul *changes* care poate fi urmat de o clauză de restricție de timp, și ea fiind opțională. Pe lângă căutarea schimbărilor, se pot căuta și autorii care au commit-uri. Structura interogărilor rămâne aceeași ca și la cele pentru schimbări, singura diferență o reprezintă înlocuirea cuvântului *changes* cu *commits*.

De asemenea, se pot scrie și interogări care caută după churn-ul autorilor. Forma acestei interogări este următoarea: cuvintele cheie *has* și *churn* urmate de o posibilă clauză de comparație prin semne și o posibilă clauză de restricție de timp, ambele clauze fiind opționale și în acest caz.

Exemple:

author: *has changes*

author: *has moreThan* 5 *commits*

author: *has exactly* 10 *commits after* 2018

author: *has churn* > 50 *since* 2017-01-17

author: *has churn* = 70

Interogările care conțin cuvintele cheie *changes*, *commits* sau *churn*, la fel ca și la cele de fișiere, sunt mai speciale și pot avea la sfârșitul lor încă o clauză în plus, și anume o clauză de restricție asupra autorilor. Clauza se referă la modificările, commit-urile sau churn-ul făcute de

autori pe anumite fișiere ale sistemului. Această clauză este opțională, iar la absența ei nu va exista nicio restricție referitoare la fișiere. Structura acestei clauze poate fi văzută în secțiunea 1.1.6.

Exemple:

```
author: has changes on "Sender.java"  
author: has changes since 2018 on every in {file: name  
startsWith "Ka"}  
author: has moreThan 5 commits on any in {file: name  
endsWith ".java"}  
author: has exactly 10 commits after 2018-10-07 on every in  
{FilesWithManyChanges}  
author: has churn >= 100 since 2017-01-17 on moreThan 3 in  
{file: size >= 650}  
author: has churn = 70 on any
```

1.4 Interogări comune

Fișierele și autorii au un tip de interogări în comun și anume interogările care au legătură cu *tag-uri*. Aceste interogări se scriu la fel și pentru fișiere și pentru autori, singura diferență este scrierea cuvântului cheie *file* sau *author*, cuvinte prin care se alege dacă interogarea să fie pentru fișiere sau pentru autori. Interogările pentru tag permit filtrarea fișierelor și a autorilor după anumite tag-uri.

Ele se pot scrie în două feluri. Prima variantă filtrează fișierele și autorii după un singur tag, iar a doua variantă permite folosirea unei liste de tag-uri. Structura interogării pentru prima variantă este următoarea: cuvântul cheie *has* urmat de semnul *hashtag* (#) și numele unui tag. La sfârșitul acestei interogări se mai poate scrie și o clauză de comparare prin semne. Dacă această clauză lipsește se vor afișa toate fișierele sau toți autorii care au acel tag cel puțin o dată. A doua variantă se scrie în felul următor: cuvântul cheie *has* urmat de unul din cuvintele *anyTag* sau *everyTag* sau de o clauză de comparare prin limbaj natural. Dacă se alege varianta cu clauza de comparare se va scrie cuvântul cheie *tags* după aceasta. Sfârșitul acestei interogări îl reprezintă cuvântul cheie *in* care este urmat de lista de tag-uri. Toate numele de tag-uri trebuie să fie precedate de semnul *hashtag* (#), iar în listă se pot scrie oricâte tag-uri, ele trebuie separate prin virgulă. Cuvântul *anyTag* reprezintă căutarea după oricare tag din listă, iar

cuvântul *everyTag* reprezintă căutarea după toate tag-urile din listă.

Exemple:

```
file: has #numeTag
author: has #numeTag > 2
file: has anyTag in [#numeTag, #altNumeTag]
author: has everyTag in [#numeTag, #altNumeTag]
file: has moreThan 2 tags in [#tag1, #tag2, #tag3]
```

1.5 Combinarea și negarea interogărilor

Exemplele de interogare din secțiunile anterioare oferă o imagine asupra posibilităților folosirii acestui limbaj obținând informațiile necesare pentru a efectua o analiză asupra sistemului software care se dorește a fi reevaluat. Interogările pentru fișiere și cele pentru autori pot fi combinate, doar cele pentru fișiere între ele și doar cele pentru autori între ele. Combinarea lor se poate face în două moduri, depinde ce se dorește. Primul mod este concatenarea lor prin cuvântul cheie *AND*, acesta impune ca toate interogările care sunt legate prin acest cuvânt să fie îndeplinite. Al doilea mod se face prin cuvântul cheie *OR*, acesta impune ca cel puțin unul din interogările care sunt legate prin acest cuvânt să fie îndeplinite.

În afară de *AND* și *OR* se mai poate folosi și cuvântul cheie *NOT*. *NOT* are ca efect negarea unei interogări. Pentru a nega o interogare se va scrie cuvântul *NOT* înaintea interogării care se dorește a fi negat.

Exemple:

```
file: name contains "Impl" AND size > 250
file: name contains "Impl" OR size > 250
file: name endsWith ".java" AND NOT size > 500
file: NOT name endsWith ".java"
```