

Part 1: SonarCloud – Static Analysis

Task 0

0.1 Specify which of the following options you have chosen. (e.g. Option 1: "Hangman2017")

Vi valgte alternativ 2.

Task 1

1.1 What does the metrics say about the code/project in whole (e.g. quality gate)?

Helhetlig så fungerer programmet, med få defekter. Det er også lav duplikasjoner

1.2 Briefly explain what the types of issues listed by the analyzer mean?

1.1.1 Bugs: det var en bug som ble funnet etter analysen ble gjort, dette var feil bruk av liste i klassen CandidateGroup.java.

1.2.2 Vulnerabilities:

her fant analysen ingen svakheter.

1.2.3 Code Smells:

Her er det listet opp 67 hvorav 3 av code smells er av kritisk grad, 39 er av major grad og 21 av minor. Analysen viste fram disse "code smells", fordi det viser at programmet kan ha mulige større feil.

Dette viser at det kan være mulige større feil i kildekoden.

1.2.4 Blocker:

analysen fant to blockerbugs den ene ligger i klassen box.java feilen her er måten arrayet er deklartert på. Den andre feilen ligger i board.java, som peker på samme feil som i box.java. Blocker bugs forhindrer sonarcloud å finne eventuelle feil senere i koden ettersom flere steder i koden er avhengig av arrayen.

1.2.5 Coverage:

analysen ga 83.5%, det vil si at de har testet gjennom det meste av koden, men de gjenværende koden ble hindret av de øvrige feilene, blockerbugs.

1.3 For the given project selected, name the folder(s) containing the highest number of LOC (Lines of code), and for the given folder? Mappen strategy har høyest kodelinjer.

1.3.1 How many bugs are there?

I mappen strategy er det kun en bug.

1.3.2 How many vulnerabilities?

Det er 0 vulnerabilities.

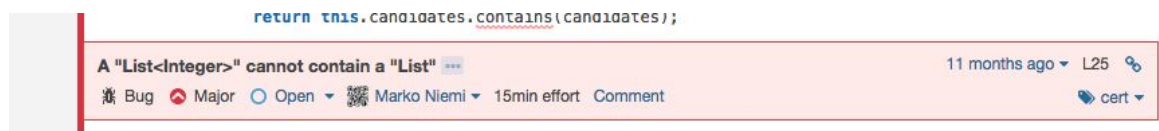
1.3.3 How many code smells?

Mappen strategy har høyest code smells på 43.

Task 2.

2.1 Select and list at least (four) 4 issues (e.g. bugs/vulnerability/code smells/. . .), and briefly explain why the analyzer has reported the given issues?

Bugs: Testen gir 1 bug, ved klassen candidateGroup.java og gir feil fordi på linje 25 står det at en liste av typen integer ikke kan inneholde enda en liste.



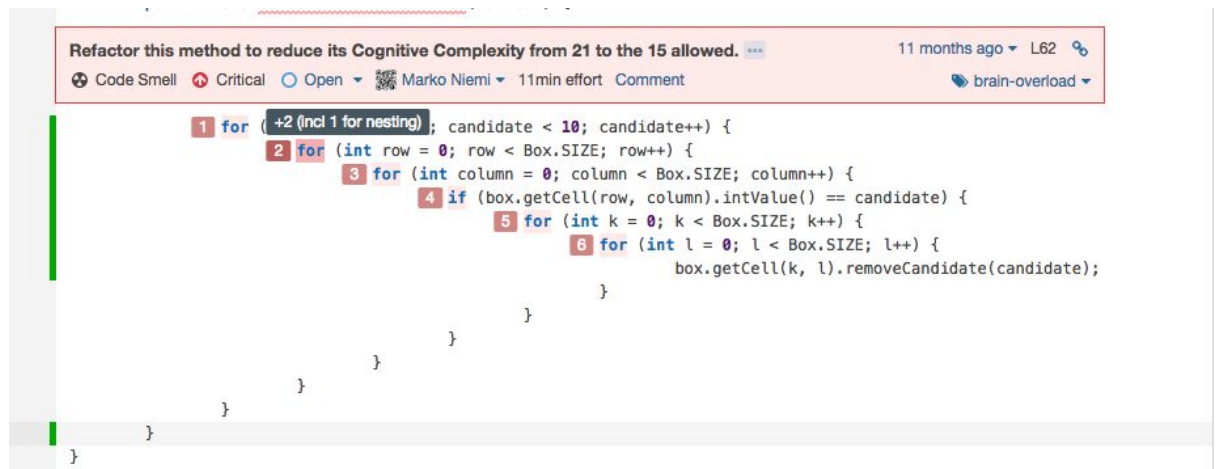
I tillegg har analysen gitt en forklaring til hvorfor det er ugunstig å skrive det på denne måten til tross for at det er mulig å kalle på “collection” metoder der man tar i bruk argumenter av feil type:

“A couple Collection methods can be called with arguments of an incorrect type, but doing so is pointless and likely the result of using the wrong argument. This rule will raise an issue when the type of the argument to List.contains or List.remove is unrelated to the type used for the list declaration.”

Code smells:

Kritisk:

Code smell-en her er at kompleksiteten i koden er altfor høy og kan muligens føre til problemer i ettertid. Her er det mange for-løkker og det blir vanskelig å forstå og håndtere hvis blir det et feil.



Cognitive Complexity is a measure of how hard the control flow of a method is to understand. Methods with high Cognitive Complexity will be difficult to maintain.

Major:

Code smell her er at i metoden "board" så har de en ubrukelig parameter ettersom de ikke tar i bruk argumentet i metoden.



Unused parameters are misleading. Whatever the values passed to such parameters, the behavior will be the same.

Compliant Solution

```
void doSomething(int a) {
    compute(a);
}
```

Blocker:

Blocker her er at deklarerer av arrayet kunne blitt gjort på en annen måte, for bedre lesbarhet. Ved å gjøre dette kan man tydelig se forskjellen mellom datatype og variabel.



Array designators should always be located on the type for better code readability. Otherwise, developers must look both at the type and the variable name to know whether or not a variable is an array.

2.2 For each of the listed issues, how would you solve the problem?

Bugs: Problemet her er at metoden sjekker om en liste inneholder lister mens man har allerede deklarerert det som en liste av typen integer. Dette vil gi false hele tiden og blir bare unødvendig.

Løsning: Enten endrer man metoden til en metode som sjekker om listen inneholder integers istedenfor eller så fjerner man metoden ettersom det ikke er noe vits i å sjekke om en liste av integers inneholder lister.

Code smells:

Kritisk: For å løse dette problemet så ville vi ha brukt færre for-løkker slik at nestingen av loopene er enklere. Dette fører også til at det er mye enklere å lese koden, samtidig blir det også bedre flyt i koden.

Løsning: Forkorte koden mere, slik at det til sammen skal være 15 nestinger.

Major: I klassen BoardToTextConverter.java, metoden appendLine så har de deklarerert en metode hvor de sender med Board board som argument, men dette tas ikke i bruk i selve metoden, dette er derfor unødvendig.

Løsning: Fjerne overflødig kode. Dette bidrar til mere oversiktlig program.

Blocker:

Feilen her er bare lesbarheten som kan mulig gi feil eller gir feil og fører til at det blokkerer videre analyse av resten av koden på grunn av at senere kode er avhengig av denne kodelinjen.

Løsning: Endre koden slik at den blir mer lesbar, istedetfor skrive det **private Cell data[][]** så skriver man heller **private Cell[][] data;**.

Task 3.

3.1 Briefly explain the following concepts:

3.1.1 False-positive (aka. False-fail result)

Det er når en test av programmet mislykkes og det resulteres på en false-positiv altså at det er ingen bugs og/eller at det fungerer helt fint funksjonelt, men man får melding om at det finnes feil.

3.1.2 False-negative (aka. False-pass result)

Det er når testing mislykkes og det resulterer på en false-negativ, i motsetning til en false-positiv så er dette mye mer problematisk ettersom det gir falskt resultat av at det er ingen feil mens det faktisk er problemer i programmet. Begge er derimot veldig tidkrevende.

3.2 Provide an example of each of the concepts.

En false-positiv kan dukke opp når det er en kodelinje som er kompleks og lang. Testingen kan mislykkes og fortelle at det er feil i denne kodelinje selv om det er ingen feil. Grunnen til at testingen mislykkes kan være på grunn av kompleksiteten til kodingen. Eller er det noe i testings-programmet som fører til at når det analyserer så utelukker noe.

En false-negativ kan dukke opp på grunn av selve testings-programmet, kanskje den ikke analyserer programmer nøyaktig nok eller det er noe som fører til at det utelukker linjer av koden. Det som er farlig med false-negativ er at man ikke vet at den er der ettersom programmet vil fortsette å kjøre ettersom den ikke finner feilen.

Part 2: Selenium IDE – Test Automation

Part 2.1 Hint: Use the "record-and-play" feature on Selenium to create test cases.

Task 1.

1.0 First, navigate to <http://demo.nopcommerce.com>

1.1 Create a new test case for the following user story:

As a new student, I want to purchase 1 MacBook and 1 phone, so that I do my homework and call my friends.

1.2 Save this test case as "test-case-1-demo". Hint: End the recording any time after the products are saved in the cart.

Task 2.

2.1 Create a new user story (NB: should be different from task [1.1]) using the following format: As a h type of user i, I want h some goal i so that h some reason i.

As a writer, I want a laptop that's light and user friendly so that I don't need to spend my time learning how to use it, but so that I can start writing as soon as possible.

2.2 Create a new test case for the previous user story (see [2.1]) .

2.3 Save the previous test case from [2.2] as "test-case-2-demo"

I denne use-casen så leter brukeren etter en laptop og vil dermed se på flere laptop på websiden og vil dermed sammenligne laptop før de kjøper en.

2.4 Create at least four (4) more unique test cases, and save each of the test cases using the same pattern (i.e. test-case-[number]-demo).

Test-case-3-demo

As a dentist, I want to purchase a phone for me, so that I can keep track of my appointments.

Test-case-4-demo

As a husband, I want to purchase some jewelry for my wife, specifically a set of necklace and bracelet, so that I can make her happy.

Test-case-5-demo

As a photographer, I want to purchase a good camera, so that I can take picture of the nature.

Test-case-6-demo

As a book-enthusiast, I want to purchase two books, so that I can have something to do in my spare time.

Task 3.

3.1 Save all the previous test cases ("test-case-1-demo", "test-case-2-demo", etc . . .) as a new test suite called "test-suite-demo". Hint: Should be six (6) test cases in total.

Task 4.

4.0 First, navigate to https://en.wikipedia.org/wiki/Main_Page

4.1 Create a test case for each the following functionalities:

4.1.1 Create a new user account

4.1.2 Login to a user account

4.1.3 Logout from a user account

4.2 Save the test cases as a test suite (i.e. one file) and call the file "test-suite-wikipedia"

4.3 For each of the three(3) test cases in [4.1], briefly explain why/why not test automation was not possible.

Test 1: Creating a new user account

Det er ikke mulig å vite hva det er brukeren kommer til å skrive inn for brukernavn, fødselsdato osv av informasjon. Det kan hende at brukeren skriver ugyldig informasjon som fører til feil. Det er vanskelig å teste med 'automation' ettersom folk er uforutsigbare. (F.eks så kan kanskje brukeren ha tilgang til symboler som programmet ikke har tilgang til og ta i bruk disse symbolene som fører til ugyldige informasjon som er satt inn.

Captcha kan endres og dermed kan man ikke putte inn samme input hver gang ettersom den forventer nye symboler.

Test 2: Login to a user account

Det er vanskelig å vite om brukeren kommer til å mislykkes eller ikke i innloggingsprosessen noe testingen ikke kan automatiseres. Brukeren kan klikke på hjelp eller søke på google etter å ha mislykket. Du kan ikke automatisere brukers tankemønstre hvor de kommer til å gå fra funksjon til funksjon etter de har logget inn osv. Det er så mye brukeren kan gjøre som uforutsigbar.

Men hvis det er kun om å logge inn så er det færre funksjoner og gjør det enklere å teste, det kan derimot fortsatt være uforutsigbarhet.

Test 3: Logout from a user account

Her er det mulig fordi det for å logge seg ut fra en brukeren så trenger man kun klikke på logg ut. Dermed kan man forvente at brukeren kommer til å klikke på denne funksjonen og teste dette ut.