Onderzoek testen

Onderzoek S3

Ruby Feller, S3-DB02

Inhoudsopgave

Waarom dit onderzoek? 2

Onderzoeksmethode 2

Onderzoeksvraag 2

Deelvragen 3

- Welke manieren van testen zijn van toepassing? 3
- Hoeveel code moet gecoverd zijn door de tests? 4
- Hoe pas je deze tests toe? 5
- Conclusie 5

Bronnen 6

Waarom dit onderzoek?

Ik doe dit onderzoek zodat ik de kwaliteit van de applicatie kan waarborgen, en de klant kan overtuigen van de kwaliteit. Dit doormiddel van diverse tests.

Onderzoeksmethode

Voor dit onderzoek word gebruik gemaakt van het DOT-framework. Dit omdat het DOT-framework een gestandaardiseerd framework is, welke de kwaliteit van het onderzoek waarborgt.
Onderstaande strategieën binnen het DOT-framework zullen toegepast worden:

Library / Desk research
Er word gebruik gemaakt van de 'Library' onderzoeksstrategie, omdat er over dit onderwerp al veel te vinden is.

Lab
Aangezien de uitkomst van het onderzoek gebruikt word voor een beroepsproduct, is er voor gekozen om de strategie 'Lab' toe te passen om bepaalde concepten te testen.

Onderzoeksvraag

Wat zijn de manieren om als ontwikkelaar de kwaliteit van een gedistribueerde webapplicatie te garanderen middels testen?

Deelvragen

Welke manieren van testen zijn van toepassing?

Aangezien dit onderzoek gaat over een gedistribueerde webapplicatie, gaat het om zowel de front-end als database en back-end.

Unit tests

Unit tests testen afzonderlijke, geïsoleerde stukjes code onafhankelijk van elkaar. Er worden diverse testcases opgesteld, en er kan gebruik gemaakt worden van “mockdata”. Het is bedoelt om de valideren dat een stukje code doet wat verwacht word, in verschillende scenario’s. (Wikipedia-bijdragers, 2022)

Integration tests

In een integratie test worden meerdere softwaremodules aan elkaar gekoppeld, en word getest of deze samenwerken zoals verwacht. Er kan bijvoorbeeld gekeken worden of er data in de database terecht komt, of dat bijvoorbeeld microservices correct samenwerken. Daarnaast kan er gekeken worden of een API-request correct afgehandeld word. (Fowler, 2018)

Functionele tests

Een functionele test focust zich op business requirements. De test kijkt of de output correct is. Ook bij deze test moeten meerdere softwarecomponenten met elkaar samenwerken. Het verschil met een integratie test is dat een integratietest niet per se naar de output kijkt, maar bijvoorbeeld alleen test of POST-request gedaan kan worden. Een functionele test kijkt wel altijd of de output correct is. (Technosoft, 2020)

End-to-end tests

In een end-to-end test word het gedrag van een gebruiker gekopieerd, en word getest of de volledige applicatie doet wat verwacht word. Dit kan gaan om het laden van een pagina, het sturen van een SMS etc. Dit type test is erg duur en kost veel tijd om te onderhouden. De test kan handig zijn voor de belangrijkste functionaliteiten, maar kleinere functionaliteiten kunnen beter getest worden door unit-en integratietests, omdat je, indien er wat fout gaat, er dan al achter zou moeten komen. (Pittet, n.d.)

Acceptatietests

Acceptatietesten testen de business requirements van het system. Ook hier word het gedrag van een gebruiker gekopieerd. Het verschil met een functionele test, is dat in een acceptatietest vanuit de business word gekeken of het systeem acceptabel is. Men kan er bijvoorbeeld achter komen dat een requirement niet volledig is gedefinieerd. Een acceptatietest word meestal niet automatisch uitgevoerd, maar bijvoorbeeld door een gebruiker. (Wikipedia-bijdragers, Acceptatietest, 2022)

Regressietest

Een regressietest word uitgevoerd na aanpassingen in de code. Er word getest of de ongewijzigde delen van de code nog steeds werken. (Wikipedia-bijdragers, Regressietest, 2022)

Front-end tests

In een front-end test worden diverse UI-flows getest. Er word bijvoorbeeld gekeken of een bepaald element op het juiste moment word ingeladen. Deze test worden vaak geautomatiseerd. (Testing: What, When and Where, n.d.)

Performance tests

In de performance test word gekeken naar de snelheid van een applicatie onder bepaalde omstandigheden. (Henson, 2019)

De testmethodes die van toepassing zijn in de applicatie zijn: integratietests voor het testen van de API-calls en database, functionele tests om te kijken of de output correct is, regressietests om te controleren of de ongewijzigde code nog werkt, acceptatietests om vast te stellen dat de applicatie voldoet aan de eisen, en front-end tests om er voor te zorgen dat de UI werkt zoals verwacht. Tot slot zijn performance tests relevant om te kijken of de website snel genoeg is.

Ik kies niet voor unittests, omdat deze in het geval van een REST-API minder relevant zijn. In een integratietest kan namelijk hetzelfde doel getest worden, maar word ook direct gekeken of het integreert met een ander deel van de applicatie. Daarnaast kies ik niet voor end-to-end test, omdat deze veel tijd eisen en lastig te onderhouden zijn.

Hoeveel code moet gecoverd zijn door de tests?

Op deze vraag is geen eenduidig antwoord, omdat er veel verschillende meningen zijn.
Het percentage van geteste code is tevens niet het meest belangrijk. Ook bij een hoog percentage, kunnen de tests namelijk slecht of onvolledig geschreven zijn, hierbij zouden dus niet alle fouten afgevangen worden.
Het is belangrijker dat de tests controleren of de requirement goed geïmplementeerd is, niet dat iedere lijn code getest word. (Pittet, What is code coverage?, n.d.)

Kortom geeft het percentage geteste code alleen aan of het stuk code getest is, niet of deze op correcte wijze getest is. Codescanners kunnen aangeven welke stukken niet getest zijn. De ontwikkelaar kan dan zelf een overweging maken of dit relevant is om te testen of niet.

Hoe pas je deze tests toe?

Back-end-IP
Voor de Spring Boot REST-API kan gebruik gemaakt worden van Junit, de meest gebruikte test library voor Java.
Voor de integratietest kan daarnaast Testcontainers gebruikt worden: een library met ondersteuning voor Junit, welke gebruik kan maken van een Docker image met daarin een database die dezelfde versie heeft als op productie. Dit zorgt ervoor dat een developer niets hoeft in te richten op zijn eigen machine, maar dat de test toch betrouwbaar is.

Front-end-IP
Voor de React front-end kan gebruik gemaakt worden van 2 libraries: Jest voor het schrijven en runnen van tests van de code en React Testing Library om de gebruikersinteractie te testen.

In mijn Java Spring Boot applicatie heb ik de integratietests toegepast, om te zien of mijn applicatie correct communiceert met de MySQL database. In onderstaande figuur is een voorbeeld van een integratie test te zien.

```
Feller,Ruby R
@Test
void When_addAssignment_Verify_Fields() {
    AssignmentDTO assignmentDTO = new AssignmentDTO();
    assignmentDTO.setTitle("test");
    assignmentDTO.setDescription("test");
    assignmentDTO.setUserId("1");
    assignmentDTO.setCreator("1");

    assignmentService.addAssignment(assignmentDTO);

    AssignmentDTO expectedAssignment = assignmentService.getById(1);

    assertEquals(assignmentDTO.getTitle(), expectedAssignment.getTitle());
    assertEquals(assignmentDTO.getDescription(), expectedAssignment.getDescription());
    assertEquals(assignmentDTO.getUserId(), expectedAssignment.getUserId());
    assertEquals(assignmentDTO.getCreator(), expectedAssignment.getCreator());

}
```

Figuur 1 integratietest om te testen of opdracht goed aangemaakt word

Conclusie

Een ontwikkelaar kan de kwaliteit van zijn gedistribueerde webapplicatie garanderen middels testen. Het percentage zegt niet per se iets over de kwaliteit, maar geeft slechts een indicatie van de hoeveelheid tests. Het is belangrijk om de juiste testmethodes te kiezen, deze zijn per applicatie verschillend. Voor mijn eigen applicatie blijken integratietests het meest toe te voegen, gezien je in een gedistribueerde meerdere modules hebt die afhankelijk van elkaar zijn.

Bronnen

Fowler, M. (2018, Januari 16). *IntegrationTest*. Martin Fowler: https://martinfowler.com/bliki/IntegrationTest.html

Henson, A. (2019, Juni 19). *Continuous Front End Website Performance Testing*. HackerNoon/Medium: https://medium.com/hackernoon/continuous-front-end-website-performance-testing-f50008beb0

Pittet, S. (n.d.). *The different types of software testing*. Atlassian: https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing

Pittet, S. (n.d.). *What is code coverage?* Atlassian: https://www.atlassian.com/continuous-delivery/software-testing/code-coverage#:~:text=With%20that%20being%20said%20it,fairly%20low%20percentage%20of%20coverage.

Technosoft. (2020, Mei 18). *Functioneel testen*. Technosoft: https://www.technosoft.nl/qa-testing/functioneel-testen

Testing: What, When and Where. (n.d.). Canvas: https://fhict.instructure.com/courses/12516/pages/testing-what-when-and-where?module_item_id=835881

Wikipedia-bijdragers. (2022, Mei 25). *Acceptatietest*. Wikipedia: https://nl.wikipedia.org/wiki/Acceptatietest

Wikipedia-bijdragers. (2022, Mei 2022). *Regressietest*. Wikipedia: https://nl.wikipedia.org/wiki/Regressietest

Wikipedia-bijdragers. (2022, oktober 26). *Unittesten*. Wikipedia: https://nl.wikipedia.org/wiki/Unittesten