Research Bronnen:

Testing algemeen:

<https://learn.microsoft.com/en-us/training/modules/visual-studio-test-concepts/>

Cursus school

Mocking in unit testing:

https://www.telerik.com/blogs/how-to-simplify-your-csharp-unit-testing-mocking-framework

<https://fakeiteasy.github.io/docs/7.3.1/>

<https://softchris.github.io/pages/dotnet-moq.html#creating-our-first-mock>

<https://www.c-sharpcorner.com/UploadFile/dacca2/fundamental-of-unit-testing-understand-mock-object-in-unit/>

<https://torvo.com.au/articles/why-mocks-are-considered-harmful>

<https://www.accenture.com/us-en/blogs/software-engineering-blog/to-mock-or-not-to-mock-is-that-even-a-question>

https://www.geeksforgeeks.org/software-testing-mock-testing/

Youtube videos 🡪 Tim Corey (Altijd goed, altijd veel te lang)

Intro to unit testing 🡪 https://www.youtube.com/watch?v=ub3P8c87cwk&ab\_channel=IAmTimCorey

Mocking in unit tests 🡪 https://www.youtube.com/watch?v=DwbYxP-etMY&ab\_channel=IAmTimCorey

https://raaaimund.github.io/tech/2019/05/07/aspnet-core-unit-testing-moq/

**Voordelen testing:**

* Kwaliteit van code verbeteren
* Bugs beperken voor ze naar productie gaan
* Helpt om betere code te schrijven, kleine testen kunnen niet uitgevoerd worden op enorm grote methodes. Hierdoor ga je er makkelijker op letten dat je meerdere kleinere methodes/functies gaat schrijven in plaats van 1 grote.
* Regressie tests, elke keer wanneer er wijzingen gebeuren aan de code kan je de testen laten runnen om te kijken of alles nog werkt zoals het zou moeten.
* Documentatie: het schrijven van tests kan helpen bij het documenteren van de code, testen beschrijven hoe de code zou moeten werken.

**Mocking in unit testing:**

Wat is het?

Fake objecten gebruiken om database operaties te imiteren of om classes te testen die een dependency hebben met een andere service, zonder dat er echt een verbinding wordt gemaakt 🡪 hiervoor dienen integratie testen/E2E testen.

Stel je wilt testen of de methode om iets te verwijderen uit een database werkt maar je wilt geen echt record verwijderen uit de database. Of stel dat de database nog niet geconfigureerd is wat dan? Mocking kan hier een antwoord op bieden

Voordelen?

Snelle en stabiele manier om database operaties te testen.

Helpt om functies te testen die een dependency hebben met andere functies die nog in development zijn.

Nadelen?

Geen integratie test

Kan refactoring moeilijker maken

Neemt simpliciteit weg

Neemt veel tijd in beslag op lange termijn

Wat kunnen we hiervoor gebruiken?

Frameworks zoals FakeItEasy / MOQ/JustMock