

ONDERZOEKSVOORSTEL

Proof of concept: De update automatiseren van Angular versie 16 naar versie 20 in de applicaties van een financiële instelling.

Bachelorproef, 2025-2026

Wauters Sander

E-mail: sander.wauters@student.hogent.be

Co-promotor: S. Beekman (Synalco, sigrid.beekman@synalco.be)

Samenvatting

Het overschakelen naar een nieuwere versie van een framework kan veel tijd in beslag nemen, vooranamelijk als de huidige applicatie meerdere versies achter loopt. Dit onderzoek heeft hierdoor als doel de update, van het web framework Angular van versie 16 naar versie 20, te automatiseren zonder de functionele of niet-functionele vereisten te schenden. Framework updates kunnen de performantie en veiligheid van de applicatie verbeteren, dus is het van belang deze uit te voeren. Deze updates zijn niet altijd even simpel om toe te passen. Functies worden aangepast of vervangen, naamgevingen veranderen, etc. In grotere codebases neemt het proces van systematisch alle veranderingen aan te brengen veel tijd in beslag. Dit onderzoek zal nagaan op welke verschillende manieren deze aanpassingen automatisch kunnen toegepast worden. Het automatisatie proces zal getest worden binnen een bedrijf dat beschikt over meerdere applicaties waarvan Angular van versie 16 naar versie 20 overgeschakel moet worden. De effectiviteit wordt beoordeeld aan de hand van het aantal veranderingen het proces correct kan uitvoeren ten opzichte van het totaal aantal uit te voeren aanpassingen. Er wordt verwacht dat 75% van alle nodige veranderingen automatisch uitgevoerd kunnen worden. Verder verwacht dit onderzoek dat het automatisatie proces uitgebreid kan worden voor gebruik bij de volgende versies van Angular mits de nodige aanpassingen.

Keuzerichting: Mobile & Enterprise development

Sleutelwoorden: Angular, Static code analysis

Inhoudsopgave

1	Inleiding	1
2	Literatuurstudie	2
3	Methodologie	2
4	Verwacht resultaat, conclusie	2
	Referenties	2

1. Inleiding

Het bedrijf Stater maakt momenteel gebruik van het web application framework Angular voor het maken van verschillende applicaties. De huidige versie van Angular die binnen Stater in gebruik is, is Angular versie 16. Op dit moment is de laatste stabiele versie van Angular, versie 20.

Stater wil graag de huidige applicaties updaten naar versie 20. **Angular<empty citation>** voorziet tooling voor een applicatie automatisch naar een nieuwe versie te updaten. Maar volgens dezelfde bron is dit gelimiteerd tot applicaties dat enkel 1 versie uiteenlopen. Dit betekent dat de update handmatig zal uitgevoerd moeten worden. Gezien de huidige schaal van de codebase zal dit veel tijd in beslag nemen. Hieruit komt de vraag: Is het mogelijk om een web applicatie in Angular versie 16 automatisch te updaten naar Angular

versie 20.

Voor deze vraag te beantwoorden worden volgende deelvragen geformuleerd:

- Wat zijn de veranderingen tussen Angular versie 16 en Angular versie 20?
- Welke van deze veranderingen kunnen automatisch toegepast worden zonder de functionele of niet-functionele vereisten van de applicatie in drang te brengen?
- Welke manieren bestaan er om code automatisch aan te passen?
- Wat is de meest geschikte manier om code automatisch aan te passen voor de applicaties van dit bedrijf?

Het automatisatie proces zal uitgevoerd worden op een Angular applicatie waarvan op voorhand geweten is hoeveel aanpassingen er moeten gebeuren. De effectiviteit het proces wordt beoordeeld aan de hand van het aantal aanpassingen automatisch uitgevoerd kunnen worden ten opzichte van het totaal aantal aanpassingen.

In de volgende sectie wordt een overzicht gegeven van de stand van zaken binnen het probleem en onderzoeksdomein. Vervolgens wordt

de methodologie van dit onderzoek beschreven. En tenslotte worden de verwachte resultaten besproken, waarin de bevindingen worden samengevat.

2. Literatuurstudie

Hier beschrijf je de *state-of-the-art* rondom je gekozen onderzoeksdomein, d.w.z. een inleidende, doorlopende tekst over het onderzoeksdomein van je bachelorproef. Je steunt daarbij heel sterk op de professionele *vakliteratuur*, en niet zozeer op populariserende teksten voor een breed publiek. Wat is de huidige stand van zaken in dit domein, en wat zijn nog eventuele open vragen (die misschien de aanleiding waren tot je onderzoeksvraag!)?

Je mag de titel van deze sectie ook aanpassen (literatuurstudie, stand van zaken, enz.). Zijn er al gelijkaardige onderzoeken gevoerd? Wat concluderen ze? Wat is het verschil met jouw onderzoek?

Verwijs bij elke introductie van een term of bewering over het domein naar de vakliteratuur, bijvoorbeeld (Hykes, 2013)! Denk zeker goed na welke werken je refereert en waarom.

Draag zorg voor correcte literatuurverwijzingen! Een bronvermelding hoort thuis *binnen* de zin waar je je op die bron baseert, dus niet er buiten! Maak meteen een verwijzing als je gebruik maakt van een bron. Doe dit dus *niet* aan het einde van een lange paragraaf. Baseer nooit teveel aansluitende tekst op eenzelfde bron.

Als je informatie over bronnen verzamelt in JabRef, zorg er dan voor dat alle nodige info aanwezig is om de bron terug te vinden (zoals uitvoerig besproken in de lessen Research Methods).

Je mag deze sectie nog verder onderverdelen in subsecties als dit de structuur van de tekst kan verduidelijken.

3. Methodologie

Hier beschrijf je hoe je van plan bent het onderzoek te voeren. Welke onderzoekstechniek ga je toepassen om elk van je onderzoeksvragen te beantwoorden? Gebruik je hiervoor literatuurstudie, interviews met belanghebbenden (bv. voor requirements-analyse), experimenten, simulaties, vergelijkende studie, risico-analyse, PoC, ...?

Valt je onderwerp onder één van de typische soorten bachelorproeven die besproken zijn in de lessen Research Methods (bv. vergelijkende studie of risico-analyse)? Zorg er dan ook voor dat we duidelijk de verschillende stappen terug vinden die we verwachten in dit soort onderzoek!

Vermijd onderzoekstechnieken die geen objectieve, meetbare resultaten kunnen opleveren. Enquêtes, bijvoorbeeld, zijn voor een bachelorproef informatica meestal **niet geschikt**. De ant-

woorden zijn eerder meningen dan feiten en in de praktijk blijkt het ook bijzonder moeilijk om voldoende respondenten te vinden. Studenten die een enquête willen voeren, hebben meestal ook geen goede definitie van de populatie, waardoor ook niet kan aangetoond worden dat eventuele resultaten representatief zijn.

Uit dit onderdeel moet duidelijk naar voren komen dat je bachelorproef ook technisch voldoende diepgang zal bevatten. Het zou niet kloppen als een bachelorproef informatica ook door bv. een student marketing zou kunnen uitgevoerd worden.

Je beschrijft ook al welke tools (hardware, software, diensten, ...) je denkt hiervoor te gebruiken of te ontwikkelen.

Probeer ook een tijdschatting te maken. Hoe lang zal je met elke fase van je onderzoek bezig zijn en wat zijn de concrete *deliverables* in elke fase?

4. Verwacht resultaat, conclusie

Hier beschrijf je welke resultaten je verwacht. Als je metingen en simulaties uitvoert, kan je hier al mock-ups maken van de grafieken samen met de verwachte conclusies. Benoem zeker al je assen en de onderdelen van de grafiek die je gaat gebruiken. Dit zorgt ervoor dat je concreet weet welk soort data je moet verzamelen en hoe je die moet meten.

Wat heeft de doelgroep van je onderzoek aan het resultaat? Op welke manier zorgt jouw bachelorproef voor een meerwaarde?

Hier beschrijf je wat je verwacht uit je onderzoek, met de motivatie waarom. Het is **niet** erg indien uit je onderzoek andere resultaten en conclusies vloeien dan dat je hier beschrijft: het is dan juist interessant om te onderzoeken waarom jouw hypothesen niet overeenkomen met de resultaten.

Referenties

Hykes, S. (2013, maart 21). *The future of Linux Containers (PyCon 2013)*. Verkregen september 1, 2016, van <https://www.youtube.com/watch?v=wW9CAH9nSLs>