

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

Samenvatting

Het overschakelen naar een nieuwere versie van een framework kan veel tijd in beslag nemen, vooranamenlijk 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
	2.1 Angular versies	2
	2.2 Automatisatie proces	2
3	Methodologie	2
4	Verwacht resultaat, conclusie	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** voor ziet tooling voor een applicatie automatisch naar een nieuwe versie te updaten. Maar volgens deze bron is dit gelimiteerd tot applicaties dat enkel 1 versie uiteenlopen. Dit betekend 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

2.1. Angular versies

De **Angular update guide** ^{<empty citation>} geeft ons een compleet overzicht van alle aanpassingen die moeten gebeuren voor de applicatie tot versie 20 te updaten. Deze updates kunnen onverdeeld worden in volgende categorieën:

- Packages updaten naar een nieuwe versie.
- Toevoegen van nieuwe configuraties.
- Vervangen van verouderde functionaliteiten.
- Hernoemen van functies en variabelen.
- Verwijderen van verouderde functies en variabelen.

2.2. Automatisatie process

De eenvoudigste manier om code in bulk aan te passen is het gebruik maken van regex gebaseerde find en replace tools. Dit is een snelle en makelijke manier om code aan te passen zonder dat er nood is aan speciale tooling. Regex is text gebaseerd en kan dus geen rekening houden met de semantiek van de programmeertaal. Als gevolg kunnen aanpassingen ongewenste gevolgen met zich meebrengen.

Een andere manier is abstract syntax tree (AST) gebaseerd refactoren. Een AST geeft informatie over de semantiek van de programmeertaal. Dit laat ons toe om een stuk code aan te passen enkel als het in een bepaalde scope zit. Voor het opstellen van een AST bestaan er verschillende open-source programmas. Een nadeel aan deze manier is dat de abstract syntax tree op de correcte manier uitgelezen moet worden om correcte aanpassingen te maken.

Nog een alternatief is om gebruik te maken van compiler tooling. Angular is gebaseerd op TypeScript en moet gecompileerd worden naar JavaScript. Compilers komen ingebouwd met de kennis van de verschillende scopes in een programmeertaal. Een nadeel aan deze aanpak is dat het uitbreiden van een compiler een complexe taak is.

Een ander optie is door Language Server Protocol (LSP) integratie. LSP's komen met programmeertaal specifieke tools zoals: code completion, syntax highlighting, het markeren van warnings en errors, etc. In het begin van 2020 zijn LSP's de "norm" geworden voor de implementatie van intelligente programmeertaal tools. Deze tools kunnen makkelijk uitgebreid worden door middel van

script. Een nadeel is dat deze aanpak gelimiteerd is aan de functies van de LSP.

Tenslotte is het mogelijk om AI tools in te zetten voor deze aanpassingen te maken. Met de recente opkomst van AI tools dat specifiek gemaakt zijn voor programmeren is het mogelijk om deze taak uit te besteden aan deze tools. Het probleem met deze aanpak is dat AI tools niet 100% betrouwbaar zijn.

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 antwoorden 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 voor 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.