

Titel van de bachelorproef.

Optionele ondertitel.

Ernst Aarden.

Scriptie voorgedragen tot het bekomen van de graad van
Professionele bachelor in de toegepaste informatica

Promotor: Dhr. F. Van Houte

Co-promotor: Mevr. S. Beeckman

Academiejaar: 2024–2025

Eerste examenperiode

Departement IT en Digitale Innovatie .

**HO
GENT**

Woord vooraf

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Samenvatting

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Inhoudsopgave

Lijst van figuren	vi
Lijst van tabellen	vii
Lijst van codefragmenten	viii
1 Inleiding	1
1.1 Probleemstelling	1
1.2 Onderzoeksvraag	2
1.3 Onderzoeksdoelstelling	2
1.4 Opzet van deze bachelorproef	3
2 Stand van zaken	4
3 Methodologie	9
4 Conclusie	11
A Onderzoeksvoorstel	13
A.1 Inleiding	13
A.2 Literatuurstudie	14
A.2.1 Uit te voeren veranderingen	14
A.2.2 Het automatisatie process	15
A.3 Methodologie	16
A.3.1 Literatuurstudie	16
A.3.2 Ontwikkeling van de proof of concept	16
A.3.3 Evaluatie	16
A.4 Verwacht resultaat, conclusie	17
Bibliografie	18

Lijst van figuren

2.1 Voorbeeld figuur.	5
-------------------------------	---

Lijst van tabellen

2.1 Voorbeeld tabel	8
-------------------------------	---

Lijst van codefragmenten

2.1	Voorbeeld codefragment	5
-----	------------------------	---

1

Inleiding

Softwareframeworks zoals Angular vereenvoudigen het maken van dynamische webapplicaties. Zoals vele software krijgt Angular geregeld updates. Deze updates komen met verschillende voordelen, zoals: nieuwe functionaliteiten, betere prestatie, bugfixes, Het toepassen van deze updates is niet altijd even vanzelfsprekend. Soms dienen nieuwe functionaliteiten als vervanging voor oudere functionaliteiten. Dit zorgt ervoor dat de code die Angular aanspreekt ook moet veranderen. Het bedrijf Stater is een end-to-end dienstverlener voor zowel hypothecaire als consumentenkredieten. Ze ondersteunen de kredietverstrekker voor de dienstverlening aan consumenten. Stater heeft intern meerdere applicaties die gebruikmaken van het Angular-framework. Specifiek gebruiken deze applicaties Angular versie 16 (v16). Stater zou graag al deze applicaties updaten naar de meest recente versie, Angular versie 20 (v20).

1.1. Probleemstelling

De sprong van 4 versies betekent wellicht dat er veel aanpassingen aan de broncode nodig zijn. Dit probleem vermeerderd zich met de grootte van de broncode en het aantal applicaties dat deze updates nodig heeft. Het manueel uitvoeren van al deze veranderingen neemt veel tijd in beslag. Dit probleem is niet eenmalig. Angular krijgt volgens Callaghan (2023) een nieuwe versie om de 6 maanden. Al dit tesamen zorgt ervoor dat de onderhoudskosten snel oplopen. De studie door Kaur en Singh (2015) beweert dat het onderhouden van een softwareproject gemiddeld 60% van de totale kostprijs in beslag neemt. Het tijdig uitvoeren van deze updates is in de praktijk niet altijd mogelijk. Buiten het onderhouden van software worden er nieuwe functies toegevoegd of wordt aan een andere applicatie gewerkt. Dit soort onderhoud kan ook niet eeuwig uitgesteld worden. Software-updates zijn volgens Vaniea en Rashidi (2016) noodzakelijk om de cyberveiligheid van een appli-

catie te garanderen.

Om deze redenen is het vereenvoudigen van het updateproces best interessant. Voor de programmeurs die de updates toepassen, vermindert de werkdruk. Voor het bedrijf Stater betekent dit dat de onderhoudstijd/-kost voor hun applicaties lager kan liggen.

1.2. Onderzoeksvraag

Op basis van de bovenstaande probleemstelling is de volgende onderzoeksvraag geformuleerd: in welke mate kan de automatisering van het updateproces van Angular v16 naar v20, bij meerdere applicaties, de onderhoudstijd voor de ontwikkelaars verlagen?

Om deze onderzoeksvraag te beantwoorden, zijn de volgende deelvragen opgesteld:

- Hoeveel veranderingen moeten uitgevoerd worden om Angular van v16 naar v20 te updaten?
- Welke manieren bestaan er om code automatisch aan te passen zonder ongewenste veranderingen uit te voeren?
- Welke manier om code automatisch aan te passen is het meest geschikt om in deze casus toe te passen?
- Wat zijn statistisch gezien de meest voorkomende problemen bij het updaten van code?

1.3. Onderzoeksdoelstelling

Om de onderzoeksvraag te beantwoorden, wordt als proof of concept een applicatie ontwikkeld die de programmeurs ondersteunt in het updateproces. In de rest van dit onderzoek zal naar deze applicatie verwezen worden als de “updater”.

Buiten de functionele requirements van de updater zal dit onderzoek proberen rekening te houden met de ruimere bedrijfscontext. Dit houdt in dat de gekozen implementatie rekening houdt met de huidige doelgroepen en de middelen/noden van het bedrijf.

Concreet betekend dit dat de updater aan de volgende criteria moet voldoen:

- De updater is van de ontwikkelaars voor de ontwikkelaars. De bedoeling is dat de persoon die de update uitvoert de updater kan instellen.
- De updater moet aanpasbaar zijn aan de uit te voeren update. Het moet kunnen gebruikt worden bij de volgende update.
- De updater mag geen nieuwe bugs introduceren. Gegeven dat de configuratie correct is, mag het geen fouten maken.

- De updater mag niet gekoppeld zijn aan Angular. Dit zorgt ervoor dat de updater zelf niet geüpdatet moet worden bij een nieuwe Angular-versie.
- De updater stuurt geen informatie door aan derde partijen. Het bedrijf bevindt zich in de financiële sector, waardoor confidentialiteit een prioriteit is.

1.4. Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie. Hier geven we een omschrijving van wat Angular is en hoe een Angular-project is opgebouwd. Verder overlopen we wat refactoring is en welke manieren reeds bestaan om dit te automatiseren.

In Hoofdstuk 3 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen. De methodologie begint met het toelichten van de gekozen refactoringstechnieken uit de literatuurstudie. Hierna volgt een korte oplijsting van welke veranderingen concreet uitgevoerd moeten worden om een Angular-applicatie van v16 naar v20 te updaten. Vervolgens wordt als proof of concept de updater uitgewerkt op basis van de gekozen technieken. Tegelijk wordt een gecontroleerde omgeving gemaakt die dient om de effectiviteit van de updater te testen. Tenslotte geven we de resultaten van de updater.

In Hoofdstuk 4, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Alle deelvragen worden beantwoord. Om het onderzoek af te ronden, worden enkele nieuwe inzichten gegeven. Deze kunnen dienen als aanzet tot verder onderzoek.

2

Stand van zaken

Dit hoofdstuk bevat je literatuurstudie. De inhoud gaat verder op de inleiding, maar zal het onderwerp van de bachelorproef *diepgaand* uitspitten. De bedoeling is dat de lezer na lezing van dit hoofdstuk helemaal op de hoogte is van de huidige stand van zaken (state-of-the-art) in het onderzoeksdomein. Iemand die niet vertrouwd is met het onderwerp, weet nu voldoende om de rest van het verhaal te kunnen volgen, zonder dat die er nog andere informatie moet over opzoeken (Pollefliet, [2011](#)).

Je verwijst bij elke bewering die je doet, vakterm die je introduceert, enz. naar je bronnen. In \LaTeX kan dat met het commando `\textcite{}` of `\autocite{}`. Als argument van het commando geef je de “sleutel” van een “record” in een bibliografische databank in het Bib \LaTeX -formaat (een tekstbestand). Als je expliciet naar de auteur verwijst in de zin (narratieve referentie), gebruik je `\textcite{}`. Soms is de auteursnaam niet expliciet een onderdeel van de zin, dan gebruik je `\autocite{}` (referentie tussen haakjes). Dit gebruik je bv. bij een citaat, of om in het bijschrift van een overgenomen afbeelding, broncode, tabel, enz. te verwijzen naar de bron. In de volgende paragraaf een voorbeeld van elk.

Knuth ([1998](#)) schreef een van de standaardwerken over sorteer- en zoekalgoritmen. Experts zijn het erover eens dat cloud computing een interessante opportuniteit vormen, zowel voor gebruikers als voor dienstverleners op vlak van informatietechnologie (Creeger, [2009](#)).

Let er ook op: het `cite`-commando voor de punt, dus binnen de zin. Je verwijst meteen naar een bron in de eerste zin die erop gebaseerd is, dus niet pas op het einde van een paragraaf.

Sed commodum posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis



Figuur 2.1: Voorbeeld van invoegen van een figuur. Zorg altijd voor een uitgebreid bijschrift dat de figuur volledig beschrijft zonder in de tekst te moeten gaan zoeken. Vergeet ook je bronvermelding niet!

```
1 import pandas as pd
2 import seaborn as sns
3
4 penguins = sns.load_dataset('penguins')
5 sns.relplot(data=penguins, x="flipper_length_mm",
              y="bill_length_mm", hue="species")
```

Codefragment 2.1: Voorbeeld van het invoegen van een codefragment.

dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetur tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed

dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl. Etiam ac leo a risus tristique nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus. Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor.

Nulla in ipsum. Praesent eros nulla, congue vitae, euismod ut, commodo a, wisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nonummy magna non leo. Sed felis erat, ullamcorper in, dictum non, ultricies ut, lectus. Proin vel arcu a odio lobortis euismod. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin ut est. Aliquam odio. Pellentesque massa turpis, cursus eu, euismod nec, tempor congue, nulla. Duis viverra gravida mauris. Cras tincidunt. Curabitur eros ligula, varius ut, pulvinar in, cursus faucibus, augue.

Nulla mattis luctus nulla. Duis commodo velit at leo. Aliquam vulputate magna et leo. Nam vestibulum ullamcorper leo. Vestibulum condimentum rutrum mauris. Donec id mauris. Morbi molestie justo et pede. Vivamus eget turpis sed nisl cursus tempor. Curabitur mollis sapien condimentum nunc. In wisi nisl, malesuada at, dignissim sit amet, lobortis in, odio. Aenean consequat arcu a ante. Pellentesque porta elit sit amet orci. Etiam at turpis nec elit ultricies imperdiet. Nulla facilisi. In hac habitasse platea dictumst. Suspendisse viverra aliquam risus. Nullam pede justo, molestie nonummy, scelerisque eu, facilisis vel, arcu.

Curabitur tellus magna, porttitor a, commodo a, commodo in, tortor. Donec interdum. Praesent scelerisque. Maecenas posuere sodales odio. Vivamus metus lacus, varius quis, imperdiet quis, rhoncus a, turpis. Etiam ligula arcu, elementum a, venenatis quis, sollicitudin sed, metus. Donec nunc pede, tincidunt in, venenatis vitae, faucibus vel, nibh. Pellentesque wisi. Nullam malesuada. Morbi ut tellus ut pede tincidunt porta. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam congue neque id dolor.

Kolom 1	Kolom 2	Kolom 3
α	β	γ
A	10.230	a
B	45.678	b
C	99.987	c

Tabel 2.1: Voorbeeld van een tabel.

Donec et nisl at wisi luctus bibendum. Nam interdum tellus ac libero. Sed sem justo, laoreet vitae, fringilla at, adipiscing ut, nibh. Maecenas non sem quis tortor eleifend fermentum. Etiam id tortor ac mauris porta vulputate. Integer porta neque vitae massa. Maecenas tempus libero a libero posuere dictum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean quis mauris sed elit commodo placerat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vivamus rhoncus tincidunt libero. Etiam elementum pretium justo. Vivamus est. Morbi a tellus eget pede tristique commodo. Nulla nisl. Vestibulum sed nisl eu sapien cursus rutrum.

Nulla non mauris vitae wisi posuere convallis. Sed eu nulla nec eros scelerisque pharetra. Nullam varius. Etiam dignissim elementum metus. Vestibulum faucibus, metus sit amet mattis rhoncus, sapien dui laoreet odio, nec ultricies nibh augue a enim. Fusce in ligula. Quisque at magna et nulla commodo consequat. Proin accumsan imperdiet sem. Nunc porta. Donec feugiat mi at justo. Phasellus facilisis ipsum quis ante. In ac elit eget ipsum pharetra faucibus. Maecenas viverra nulla in massa.

Nulla ac nisl. Nullam urna nulla, ullamcorper in, interdum sit amet, gravida ut, risus. Aenean ac enim. In luctus. Phasellus eu quam vitae turpis viverra pellentesque. Duis feugiat felis ut enim. Phasellus pharetra, sem id porttitor sodales, magna nunc aliquet nibh, nec blandit nisl mauris at pede. Suspendisse risus risus, lobortis eget, semper at, imperdiet sit amet, quam. Quisque scelerisque dapibus nibh. Nam enim. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ut metus. Ut metus justo, auctor at, ultrices eu, sagittis ut, purus. Aliquam aliquam.

3

Methodologie

Etiam pede massa, dapibus vitae, rhoncus in, placerat posuere, odio. Vestibulum luctus commodo lacus. Morbi lacus dui, tempor sed, euismod eget, condimentum at, tortor. Phasellus aliquet odio ac lacus tempor faucibus. Praesent sed sem. Praesent iaculis. Cras rhoncus tellus sed justo ullamcorper sagittis. Donec quis orci. Sed ut tortor quis tellus euismod tincidunt. Suspendisse congue nisl eu elit. Aliquam tortor diam, tempus id, tristique eget, sodales vel, nulla. Praesent tellus mi, condimentum sed, viverra at, consectetur quis, lectus. In auctor vehicula orci. Sed pede sapien, euismod in, suscipit in, pharetra placerat, metus. Vivamus commodo dui non odio. Donec et felis.

Etiam suscipit aliquam arcu. Aliquam sit amet est ac purus bibendum congue. Sed in eros. Morbi non orci. Pellentesque mattis lacinia elit. Fusce molestie velit in ligula. Nullam et orci vitae nibh vulputate auctor. Aliquam eget purus. Nulla auctor wisi sed ipsum. Morbi porttitor tellus ac enim. Fusce ornare. Proin ipsum enim, tincidunt in, ornare venenatis, molestie a, augue. Donec vel pede in lacus sagittis porta. Sed hendrerit ipsum quis nisl. Suspendisse quis massa ac nibh pretium cursus. Sed sodales. Nam eu neque quis pede dignissim ornare. Maecenas eu purus ac urna tincidunt congue.

Donec et nisl id sapien blandit mattis. Aenean dictum odio sit amet risus. Morbi purus. Nulla a est sit amet purus venenatis iaculis. Vivamus viverra purus vel magna. Donec in justo sed odio malesuada dapibus. Nunc ultrices aliquam nunc. Vivamus facilisis pellentesque velit. Nulla nunc velit, vulputate dapibus, vulputate id, mattis ac, justo. Nam mattis elit dapibus purus. Quisque enim risus, congue non, elementum ut, mattis quis, sem. Quisque elit.

Maecenas non massa. Vestibulum pharetra nulla at lorem. Duis quis quam id lacus dapibus interdum. Nulla lorem. Donec ut ante quis dolor bibendum condimentum. Etiam egestas tortor vitae lacus. Praesent cursus. Mauris bibendum pede at elit. Morbi et felis a lectus interdum facilisis. Sed suscipit gravida turpis. Nulla at

lectus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Praesent nonummy luctus nibh. Proin turpis nunc, congue eu, egestas ut, fringilla at, tellus. In hac habitasse platea dictumst.

Vivamus eu tellus sed tellus consequat suscipit. Nam orci orci, malesuada id, gravida nec, ultricies vitae, erat. Donec risus turpis, luctus sit amet, interdum quis, porta sed, ipsum. Suspendisse condimentum, tortor at egestas posuere, neque metus tempor orci, et tincidunt urna nunc a purus. Sed facilisis blandit tellus. Nunc risus sem, suscipit nec, eleifend quis, cursus quis, libero. Curabitur et dolor. Sed vitae sem. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Maecenas ante. Duis ullamcorper enim. Donec tristique enim eu leo. Nullam molestie elit eu dolor. Nullam bibendum, turpis vitae tristique gravida, quam sapien tempor lectus, quis pretium tellus purus ac quam. Nulla facilisi.

4

Conclusie

Curabitur nunc magna, posuere eget, venenatis eu, vehicula ac, velit. Aenean ornare, massa a accumsan pulvinar, quam lorem laoreet purus, eu sodales magna risus molestie lorem. Nunc erat velit, hendrerit quis, malesuada ut, aliquam vitae, wisi. Sed posuere. Suspendisse ipsum arcu, scelerisque nec, aliquam eu, molestie tincidunt, justo. Phasellus iaculis. Sed posuere lorem non ipsum. Pellentesque dapibus. Suspendisse quam libero, laoreet a, tincidunt eget, consequat at, est. Nullam ut lectus non enim consequat facilisis. Mauris leo. Quisque pede ligula, auctor vel, pellentesque vel, posuere id, turpis. Cras ipsum sem, cursus et, facilisis ut, tempus euismod, quam. Suspendisse tristique dolor eu orci. Mauris mattis. Aenean semper. Vivamus tortor magna, facilisis id, varius mattis, hendrerit in, justo. Integer purus.

Vivamus adipiscing. Curabitur imperdiet tempus turpis. Vivamus sapien dolor, congue venenatis, euismod eget, porta rhoncus, magna. Proin condimentum pretium enim. Fusce fringilla, libero et venenatis facilisis, eros enim cursus arcu, vitae facilisis odio augue vitae orci. Aliquam varius nibh ut odio. Sed condimentum condimentum nunc. Pellentesque eget massa. Pellentesque quis mauris. Donec ut ligula ac pede pulvinar lobortis. Pellentesque euismod. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent elit. Ut laoreet ornare est. Phasellus gravida vulputate nulla. Donec sit amet arcu ut sem tempor malesuada. Praesent hendrerit augue in urna. Proin enim ante, ornare vel, consequat ut, blandit in, justo. Donec felis elit, dignissim sed, sagittis ut, ullamcorper a, nulla. Aenean pharetra vulputate odio.

Quisque enim. Proin velit neque, tristique eu, eleifend eget, vestibulum nec, lacus. Vivamus odio. Duis odio urna, vehicula in, elementum aliquam, aliquet laoreet, tellus. Sed velit. Sed vel mi ac elit aliquet interdum. Etiam sapien neque, convallis et, aliquet vel, auctor non, arcu. Aliquam suscipit aliquam lectus. Proin tincidunt magna sed wisi. Integer blandit lacus ut lorem. Sed luctus justo sed enim.

Morbi malesuada hendrerit dui. Nunc mauris leo, dapibus sit amet, vestibulum et, commodo id, est. Pellentesque purus. Pellentesque tristique, nunc ac pulvinar adipiscing, justo eros consequat lectus, sit amet posuere lectus neque vel augue. Cras consectetur libero ac eros. Ut eget massa. Fusce sit amet enim eleifend sem dictum auctor. In eget risus luctus wisi convallis pulvinar. Vivamus sapien risus, tempor in, viverra in, aliquet pellentesque, eros. Aliquam euismod libero a sem. Nunc velit augue, scelerisque dignissim, lobortis et, aliquam in, risus. In eu eros. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Curabitur vulputate elit viverra augue. Mauris fringilla, tortor sit amet malesuada mollis, sapien mi dapibus odio, ac imperdiet ligula enim eget nisl. Quisque vitae pede a pede aliquet suscipit. Phasellus tellus pede, viverra vestibulum, gravida id, laoreet in, justo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Integer commodo luctus lectus. Mauris justo. Duis varius eros. Sed quam. Cras lacus eros, rutrum eget, varius quis, convallis iaculis, velit. Mauris imperdiet, metus at tristique venenatis, purus neque pellentesque mauris, a ultrices elit lacus nec tortor. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent malesuada. Nam lacus lectus, auctor sit amet, malesuada vel, elementum eget, metus. Duis neque pede, facilisis eget, egestas elementum, nonummy id, neque.



Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

A.1. Inleiding

Het bedrijf Stater is een end-to-end dienstverlener voor zowel hypothecaire als consumentenkredieten. Ze ondersteunen de kredietverstrekker voor de dienstverlening aan consumenten. Binnen het bedrijf zijn er verschillende applicaties die gebruikmaken van het Angular webframework. Angular is een open-source front-end framework, ontwikkeling door Google, gebaseerd op de TypeScript programmeertaal voor de ontwikkeling van dynamische webapplicaties (Cincovic e.a., [2019](#)). Momenteel is Angular versie 20 (v20) de meest recente stabiele versie. Binnen Stater maken de applicaties gebruik van Angular versie 16 (v16). Het bedrijf is van plan de applicaties te updaten naar de recentste versie, Angular v20.

De updates niet uitvoeren is geen optie. Volgens de studie door Vaniea en Ras-hidi ([2016](#)) zijn software-updates nodig, omdat het nieuwe functionaliteiten introduceert, de performantie verbetert en verzekert dat de software compatibel blijft met andere nieuwe software. Verder omschrijft deze bron dat het up-to-date houden van software cruciaal is om de cyberveiligheid te garanderen.

Het grote verschil in versies zorgt ervoor dat het updaten van alle applicaties veel tijd in beslag neemt. Dit is geen eenmalig probleem; volgens Callaghan ([2023](#)) krijgt Angular een nieuwe versie om de 6 maanden. De studie door Kaur en Singh ([2015](#)) beweert dat het onderhouden van een softwareproject gemiddeld 60% van de kostprijs in beslag neemt. Een manier om de tijd voor software-onderhoud in te korten is daarom best interessant. Hieruit komt de vraag: in welke mate kan de automatisering van het updateproces van Angular v16 naar v20, bij meerdere

applicaties, de onderhoudstijd voor de ontwikkelaars verlagen? Om deze vraag te beantwoorden zijn de volgende deelvragen geformuleerd:

- Hoeveel veranderingen moeten uitgevoerd worden om Angular van v16 naar v20 te updaten?
- Welke manieren bestaan om code automatisch aan te passen zonder ongewenste veranderingen uit te voeren?
- Welke manier om code automatisch aan te passen is het meest geschikt om in deze casus toe te passen?
- Wat zijn statistisch gezien de meest voorkomende problemen bij het updaten van code?

Gedurende dit onderzoek zal als proof of concept een applicatie ontwikkeld worden die een softwareproject in Angular v16 automatisch updatet naar Angular v20. In de rest van dit document wordt naar deze applicatie verwezen als de “updater”. De updater doorloopt de broncode van een applicatie en maakt een olijsting van alle nodige aanpassingen en tracht de aanpassing zelf uit te voeren indien mogelijk. Het doelpubliek van de updater zijn de personen die anders deze aanpassingen aan de broncode manueel uitvoeren. De effectiviteit van de updater wordt bepaald aan het aantal gedetecteerde en opgeloste aanpassingen tegenover het totaal van de nodige aanpassingen.

In de volgende sectie wordt een kort overzicht gegeven van de huidige stand van zaken binnen het probleem- en oplossingsdomein. Hierna volgt een beschrijving van de methodologie, waar de werking en evaluatie van de updater in meer detail beschreven is. Tenslotte worden de verwachte resultaten besproken, waarin een inschatting wordt gegeven van de bevindingen van het onderzoek.

A.2. Literatuurstudie

A.2.1. Uit te voeren veranderingen

De Angular update guide door Google (2025) geeft een uitgebreid overzicht van alle aanpassingen die nodig zijn om een Angular-applicatie van v16 naar v20 te updaten. Uit deze bron blijkt dat in totaal 80 verschillende stappen uitgevoerd moeten worden. Deze stappen gaan van het uitvoeren van commando's tot verschillende aanpassingen aan de code.

Zoals omschreven in de studie door Cincović en Punt (2020), is de code in Angular applicaties onderverdeeld in 3 verschillende soorten bestanden:

- TypeScript-bestanden die de bedrijfslogica bevatten.
- HTML-bestanden die de structuur van de user interface (UI) omschrijven.
- CSS-bestanden die de visuele representatie van de UI omschrijven.

De studie door Di Penta e.a. (2020) onderzoekt welke veranderingen aan code de meeste kans hebben om nieuwe bugs te introduceren. Deze studie maakt het mogelijk om een geïnformeerde inschatting te maken van welke aanpassingen geautomatiseerd kunnen worden zonder ongewenste bijwerkingen te introduceren.

A.2.2. Het automatisatie proces

Eén van de simpelste manieren om code in bulk aan te passen is het gebruikmaken van zoek- en vervangfuncties gebaseerd op text of reguliere expressies (Regex). De studie van Michael e.a. (2019) omschrijft Regex als een sequentie van karakters die een patroon in een tekst omschrijft. Uit dezelfde studie blijken enkele problemen bij de implementatie van Regex, namelijk dat het moeilijk leesbaar, vindbaar, valideerbaar en documenteerbaar is. Verder kan Regex geen rekening houden met de semantiek van de programmeertaal, aangezien het enkel op tekst gebaseerd is. Om met de semantiek van de programmeertaal rekening te houden, kan gebruikgemaakt worden van een abstract syntax tree. Zoals omschreven door Sun e.a. (2023), een abstract syntax tree is een datastructuur die de broncode van een applicatie illustreert en rekening houdt met de syntax en semantiek van de programmeertaal. In combinatie met zoek- en vervangfuncties laat dit toe om een stuk code aan te passen, enkel als het in een bepaalde context zit.

Herinner dat Angular gebaseerd is op TypeScript. Een bestaande tool voor TypeScript die gebruikmaakt van een abstract syntax tree is de TypeScript Compiler API. De studie door Reid e.a. (2023) onderzoekt hoe de TypeScript Compiler API gebruikt kan worden voor het corrigeren van foutieve codefragmenten. Deze studie raadt aan om de TypeScript Compiler API te gebruiken voor statische code-analyse vanwege de effectiviteit, accuraatheid en mogelijkheid om foutieve code te detecteren.

Een alternatief voor de TypeScript Compiler API dat ook gebruikmaakt van een abstract syntax tree is het Angular Language Server Protocol (LSP). Het LSP, als omschreven door Bork en Langer (2023), is een open protocol voor gebruik in verschillende code-editors of integrated development environments (IDEs) dat programmeertaal-specifieke functies voorziet zoals: automatisch code aanvullen en code-diagnostics. Dezelfde bron omschrijft LSP's als het de facto standaardprotocol voor de implementatie van deze functies in IDE's.

Tenslotte is het mogelijk om artificiële intelligentie in te zetten om deze aanpassingen te maken. Met de recente opkomst van AI-tools die specifiek gemaakt zijn voor programmeren, is het mogelijk om deze taak uit te besteden aan AI. Er zijn echter problemen met deze aanpak voor deze casus. Uit de studie door Hodovychenko en Kurinko (2025) blijkt dat AI-gedreven tools een gebrek hebben aan transparantie en risico lopen de semantiek van de programmeertaal in de loop van de tijd fout te interpreteren. Verder maakt deze studie de bewering dat voor het maken van dit soort AI-tools er nood is aan een grote hoeveelheid betrouwbare trainingsdata.

Het bemachtigen van deze data is problematisch, vooral als het gaat om code die gebruikmaakt van de allernieuwste updates.

A.3. Methodologie

A.3.1. Literatuurstudie

Dit onderzoek start met een uitgebreide literatuurstudie naar de verschillen tussen Angular v16 en Angular v20. De nodige veranderingen worden opgelijst en onderverdeeld in verschillende categorieën. Deze oplijsting bepaalt de capaciteit van de updater. Verder geeft dit een beter overzicht van welke aanpassingen al dan niet geschikt zijn voor automatisatie. Tenslotte zal deze oplijsting dienen als maatstaf om de effectiviteit van de updater op te meten.

Vervolgens wordt onderzocht wat de verschillende manieren zijn om automatisch code te updaten. Eén of meerdere manieren worden verkozen om te implementeren op basis van de volgende criteria:

- Complexiteit van de implementatie. Een voorkeur wordt gegeven aan het gebruiken van bestaande tools boven het ontwikkelen van nieuwe algoritmes.
- Betrouwbaarheid van de output. Zolang de input hetzelfde blijft, mag de output niet veranderen.
- Gebruiksvriendelijkheid. Het moet bruikbaar zijn voor de persoon die normaal manueel de applicaties updatet.

A.3.2. Ontwikkeling van de proof of concept

In deze fase van het onderzoek zal de updater ontwikkeld worden op basis van de voorafgaande literatuurstudie.

De updater heeft als minimum de volgende twee functies: het detecteren van code die aangepast moet worden en de aanpassingen uitvoeren indien mogelijk. Het detecteren van de code speelt een dubbele rol. In eerste instantie is het nodig om de aanpassing op de correcte plaats uit te voeren. Indien de aanpassing niet geautomatiseerd kan worden, zorgt het voor een overzicht van waar alle nodige aanpassingen gemaakt moeten worden.

De updater wordt in een gecontroleerde omgeving getest om de stabiliteit te verzekeren. Deze gecontroleerde omgeving bestaat uit een testapplicatie gemaakt in Angular v16. De testapplicatie bevat een codefragment voor elke vooraf geïdentificeerde stap in het updateproces.

A.3.3. Evaluatie

De effectiviteit van de updater wordt bepaald aan het aantal gedetecteerde en opgeloste aanpassingen tegenover het totaal aantal aanpassingen.

Deze meting wordt uitgevoerd op de gecontroleerde omgeving die gemaakt is in de proof of concept. Dit geeft een totaalresultaat voor alle aanpassingen die theoretisch nodig zijn.

A.4. Verwacht resultaat, conclusie

Op basis van de literatuurstudie en de gehanteerde methodologie verwacht dit onderzoek dat minstens 65% van alle nodige aanpassingen automatisch uitgevoerd kan worden.

Het automatisatieproces zal naar verwachting de nodige tijd voor de applicaties te updaten verminderen. De totale hoeveelheid tijd die in werkelijkheid bespaard wordt, is afhankelijk van verschillende factoren: de ervaring van de programmeur, hun kennis van de broncode, de grootte van de applicaties, Deze oplossing zal wellicht meer tijd in beslag nemen als enkel één kleine applicatie geüpdatet moet worden.

Dit onderzoek tracht de beste methode te implementeren voor deze casus op basis van gekende literatuur. Echter, kan het interessant zijn om andere manieren te implementeren en te vergelijken. Verder onderzoek kan uitgevoerd worden naar de performantie, accuraatheid en complexiteit van de verschillende implementaties besproken in de literatuurstudie.

Bibliografie

- Bork, D., & Langer, P. (2023). Language server protocol: An introduction to the protocol, its use, and adoption for web modeling tools. *Enterprise Modelling and Information Systems Architectures (EMISAJ)*, 18, 9–1. <https://doi.org/10.18417/emisa.18.9>
- Callaghan, M. D. (2023). Upgrading Angular. In *Angular for Business: Awaken the Advocate Within and Become the Angular Expert at Work* (pp. 95–118). Springer. https://doi.org/10.1007/978-1-4842-9609-7_8
- Cincovic, J., Delcev, S., & Draskovic, D. (2019). Architecture of web applications based on Angular Framework: A Case Study. *methodology*, 7(7), 254–259. <https://www.eventiotic.com/eventiotic/files/Papers/URL/df6b5054-816e-4bee-b983-663fb87be2cd.pdf>
- Cincović, J., & Punt, M. (2020). Comparison: Angular vs. React vs. Vue. Which framework is the best choice? *Zdravković, M., Konjović, Z., Trajanović, M.(Eds.) ICIST 2020 Proceedings*, 250–255. <https://www.eventiotic.com/eventiotic/files/Papers/URL/50173409-699e-4b17-8edb-9764ecc53160.pdf>
- Creeger, M. (2009). CTO Roundtable: Cloud Computing. *Communications of the ACM*, 52(8), 50–56.
- Di Penta, M., Bavota, G., & Zampetti, F. (2020). On the relationship between refactoring actions and bugs: a differentiated replication. *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 556–567. <https://doi.org/10.1145/3368089.3409695>
- Google. (2025). *Angular update guide*. Verkregen september 8, 2025, van <https://angular.dev/update-guide?v=16.0-20.0>
- Hodovychenko, M. A. H. M. A., & Kurinko, D. D. K. D. D. (2025). Analysis of existing approaches to automated refactoring of object-oriented software systems. *Вісник сучасних інформаційних технологій*, 8(2), 179–196. <https://doi.org/10.15276/hait.08.2025.11>
- Kaur, U., & Singh, G. (2015). A review on software maintenance issues and how to reduce maintenance efforts. *International Journal of Computer Applications*, 118(1), 6–11. <https://doi.org/10.5120/20707-3021>
- Knuth, D. E. (1998). *The art of computer programming, volume 3: (2nd ed.) sorting and searching*. Addison Wesley Longman Publishing Co., Inc.

- Michael, L. G., Donohue, J., Davis, J. C., Lee, D., & Servant, F. (2019). Regexes are hard: Decision-making, difficulties, and risks in programming regular expressions. *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 415–426. <https://doi.org/10.1109/ASE.2019.00047>
- Pollefliet, L. (2011). *Schrijven van verslag tot eindwerk: do's en don'ts*. Academia Press.
- Reid, B., Treude, C., & Wagner, M. (2023). Using the TypeScript compiler to fix erroneous Node.js snippets. *2023 IEEE 23rd International Working Conference on Source Code Analysis and Manipulation (SCAM)*, 220–230. <https://doi.org/10.1109/SCAM59687.2023.00031>
- Sun, W., Fang, C., Miao, Y., You, Y., Yuan, M., Chen, Y., Zhang, Q., Guo, A., Chen, X., Liu, Y., e.a. (2023). Abstract syntax tree for programming language understanding and representation: How far are we? *arXiv preprint arXiv:2312.00413*. <https://doi.org/10.48550/arXiv.2312.00413>
- Vania, K., & Rashidi, Y. (2016). Tales of software updates: The process of updating software. *Proceedings of the 2016 chi conference on human factors in computing systems*, 3215–3226. <https://doi.org/10.1145/2858036.2858303>