

# Proof of concept: De update automatiseren van Angular versie 16 naar versie 20 in de applicaties van een end-to-end kredietdienstverlener.

Wauters Sander  
sander.wauters@student.hogent.be  
Promotor: Irina Malfait  
Co-promotor: Peter De Seranno (Stater)

Hogeschool Gent, Arbeidstraat 14, 9300 Aalst  
Keuzerichting: Mobile & Enterprise development  
Sleutelwoorden: Angular, TypeScript, Static code analysis, Automatisatie

## Samenvatting

Applicatie ontwikkeld om het updateproces van Angular-applicaties te ondersteunen. De aanpak werd getest door een Angular-applicatie te upgraden van v16 naar v20. De evaluatie toont aan dat meer dan 25% van alle noodzakelijke aanpassingen automatiseerbaar is. Deze aanpak is bruikbaar voor het refactoren van verschillende TypeScript-applicaties. De applicatie gebruikt de TypeScript-compiler voor een betrouwbare foutdetectie.

## Introductie

Het Angular-framework vereenvoudigt het ontwikkelingsproces voor het bouwen van dynamische webapplicaties. Zoals bij de meeste software ontvangt Angular regelmatig updates, die noodzakelijk zijn om onder andere de cyberveiligheid te verbeteren. Het toepassen van dergelijke updates is echter niet altijd vanzelfsprekend.

Angular verwijdert verouderde functionaliteiten uit het framework, waardoor aanpassingen aan de broncode nodig zijn. Dit soort updates vindt plaats om de 6 maanden. Wanneer er meerdere enterprise-applicaties betrokken zijn, kan de benodigde tijd voor dit updateproces snel oplopen. Het bedrijf Stater ondervindt dit probleem. Stater is een end-to-end dienstverlener voor zowel hypothecaire als consumentenkredieten. Ze willen al hun applicaties updaten van Angular v16 naar v20. Deze sprong van 4 versies vereist aanzienlijke aanpassingen aan de broncode.

## Methodologie

Dit onderzoek begon met een literatuurstudie naar bekende methoden om code automatisch aan te passen. Er werd besloten een *updater* te ontwikkelen op basis van de TypeScript Compiler API. Door de TypeScript-compiler programmatisch aan te spreken en te koppelen aan zoek- en vervangfuncties, werd het mogelijk om code gericht op te sporen en aan te passen.

Een collectie helperfuncties is ontwikkeld om de implementatie te vereenvoudigen. Dankzij de programmatische opzet biedt de updater een hoge van flexibiliteit. De updater is geïmplementeerd als een command-lineapplicatie, waardoor deze voor meerdere projecten kan worden gebruikt. Het volgende codefragment toont hoe alle instanties van een functie gericht kunnen worden hernoemd doorheen meerdere applicaties.

```
1 const project = loadProject();
2 project.getSourceFiles().forEach((file) =>
3   findNodes(
4     file,
5     (node) =>
6       deepestInstanceOf(node, "async") &&
7       node.getKind() !== SyntaxKind.AsyncKeyword,
8     (node) => node.replaceWithText("waitForAsync"),
9   ),
10 );
11 await saveProject(project);
```

## Onderzoeksvragen

In welke mate kan de automatisering van het updateproces van Angular v16 naar v20, bij meerdere applicaties, de onderhoudstijd voor de ontwikkelaars verlagen?

Deelvragen:

- Welke 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?

## Conclusies

De testresultaten tonen aan dat de updater een meerwaarde biedt bij de ondersteuning van het updateproces. Ongeveer een kwart van alle aanpassingen kan automatisch worden uitgevoerd. De gekozen manier van werken kan eenvoudig overgenomen worden voor toekomstige updates. Bovendien is de updater niet beperkt tot het uitvoeren van Angular-updates. Deze werkwijze kan ook worden toegepast voor meer algemene refactorings. Daarnaast kan de updater worden ingezet voor andere TypeScript-applicaties.

Categorie	Totaal		TypeScript		Testen		Syntax		Semantiek	
#Stappen	80	100,00%	50	100,00%	16	100,00%	24	100,00%	38	100,00%
Automatiseerbaar										
Volledig	22	27,50%	10	20,00%	3	18,75%	9	37,50%	1	2,63%
Gedeeltelijk	8	10,00%	8	16,00%	0	0,00%	5	20,83%	5	13,16%
Niet	50	62,50%	32	64,00%	13	81,25%	10	41,67%	32	84,21%
Detecteerbaar										
Volledig	24	30,00%	24	48,00%	6	37,50%	12	50,00%	13	34,21%
Gedeeltelijk	5	6,25%	5	10,00%	0	0,00%	2	8,33%	4	10,53%
Niet	51	63,75%	21	42,00%	10	62,50%	10	41,67%	21	55,26%

Resultaten van de updater om een Angular-applicatie te updaten van v16 naar v20.

## Toekomstig onderzoek

Dit onderzoek focust op één van de vele automatisatietechnieken. Een vergelijking van verschillende technieken en hun toepasbaarheid in andere casussen kan interessant zijn voor toekomstig onderzoek. Zo kan worden onderzocht in welke situaties een integratie met een language server of met AI gepast is.

Tijdens het schrijven van dit onderzoek werd Angular versie 21 uitgebracht. Deze versie introduceert nieuwe tools om AI beter te integreren in het ontwikkelingsproces. Voornamelijk beweert het Angular-team dat het AI toelaat om de nieuwste functionaliteiten te gebruiken. Hoewel deze tools momenteel nog experimenteel zijn, tonen ze potentieel voor verdere automatisering.