

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

Samenvatting

Applicatie ontwerpen voor een Angular applicatie van v16 naar v20 te updaten.

Keuzerichting: Mobile & Enterprise development

Sleutelwoorden: Angular, Static code analysis, Automatisatie

Broncode: <https://github.com/Sander-Wauters/hogent-2526-thesis-SanderWauters>

1. Introductie

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

Angular verwijdert verouderde functionaliteiten uit het framework. Daardoor moet de broncode van Angular-applicaties aangepast worden. Dit type updates vindt om de 6 maanden plaats. Bij meerdere enterprise-applicaties kan de benodigde tijd voor dit updateproces snel oplopen. Het bedrijf Stater ervaart dit probleem. Stater is een end-to-end dienstverlener voor zowel hypothecaire als consumentenkredieten. Zij willen al hun applicaties updaten van Angular v16 naar v20. De sprong van vier versies betekent dat er vermoedelijk veel aanpassingen in de broncode nodig zijn.

2. Methodologie

3. Onderzoeksvragen

Hoofdvraag:

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

Deelvragen:

- 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?

4. Conclusies

De testresultaten tonen aan dat de updaten een meerwaarde biedt in de ondersteuning van het updateproces. Ondanks dat het verwachte resultaat niet bereikt is, was het nog steeds mogelijk om een vierde van alle aanpassingen automatisch uit te voeren. De hoge flexibiliteit van deze aanpak maakt het mogelijk voor de updaten te herconfigureren voor toekomstige updates. Bovendien is de updaten niet gelimiteerd aan het uitvoeren van Angular-updates. Dezelfde manier van werken kan toegepast worden om meer algemene refactoringen uit te voeren. Verder kan men de updaten configureren om op andere TypeScript-applicaties te werken.

Categorie	Totaal
#Stappen	80 100,00%
Automatiseerbaar	
Volledig	22 27,50%
Deels	8 10,00%
Niet	50 62,50%
Detecteerbaar	
Volledig	24 30,00%
Deels	5 6,25%
Niet	51 63,75%

Tabel 1: Resultaten van de updaten uitgevoerd op de testomgeving.

5. Toekomstig onderzoek

In de stand van zaken hebben we verschillende automatisatietechnieken besproken. Een vergelijking van deze technieken voor toepassing in andere casussen kan best interessant zijn. Wanneer zou een integratie met een language server, of met AI, gepast zijn bijvoorbeeld?

Tijdens het schrijven van dit onderzoek is Angular v21 uitgekomen. Deze versie komt met nieuwe tools om AI beter te integreren in het ontwikkelingsproces. Voornamelijk beweren ze dat het AI toelaat om de nieuwste functionaliteiten te gebruiken. Dit was één van de redenen dat AI niet gekozen werd in dit onderzoek. Deze tools zijn momenteel nog experimenteel, maar kunnen veelbelovend zijn.