

# End to End Testing Framework voor de Nieuwe Front-End van Colruyt Group's Checkoutsysteem: Vergelijkende Studie & Proof of Concept

## Onderzoeksvoorstel Bachelorproef 2019-2020

Pieter-Jan Van Damme<sup>1</sup>

### Samenvatting

Colruyt Group is bezig met de ontwikkeling van een nieuwe front-end voor haar checkoutsysteem. Om de stabiliteit van het systeem op lange termijn te kunnen garanderen, is het belangrijk dat dit systeem ondersteund wordt door een uitgebreid testsysteem. De eerste stap om dit te realiseren, is het kiezen van een geschikt testframework. Dit document is een voorstel tot het onderzoeken van de geschiktheid van een reeks bestaande testframeworks voor deze use case, dit als Bachelorproef van de auteur. Het is waarschijnlijk dat elk testframework zijn unieke sterktes en zwaktes zal demonstreren. Op basis van de specifieke situatie van Colruyt Group zal de auteur dan een gefundeerd advies formuleren.

### Sleutelwoorden

Onderzoeksdomein. Webapplicatieontwikkeling — Angular — Testautomatisatie

### Co-promotor

Koen Stockman<sup>2</sup> (Colruyt Group)

Contact: <sup>1</sup> pieterjan.vandamme.p4157@student.hogent.be; <sup>2</sup> koen.stockman@colruytgroup.com;

## Inhoudsopgave

1	Introductie	1
2	State-of-the-art	1
3	Methodologie	2
4	Verwachte resultaten	2
5	Verwachte conclusies	2
	Referenties	2

## 1. Introductie

Het correct én efficiënt verwerken van klantenaankopen is een cruciaal proces binnen de retailsector. Om die reden investeert Colruyt Group momenteel in nieuwe touchscreens voor haar voornaamste kassasysteem, alsook een herwerking van de checkoutsoftware.

Hoewel de ontwikkeling van deze software met rasse schreden vooruit gaat, werd er tot op heden maar weinig geïnvesteerd in (geautomatiseerde) e2e (end to end, oftewel een simulatie van een echt gebruikersscenario waarbij alle componenten van het systeem getest worden softwaretestinghelp.com, 2019) testing. Nochtans is de stabiliteit van het kassasysteem een onbetwistbare noodzaak voor groep.

Bovendien wordt de software continu uitgebreid met nieuwe functionaliteit en moet het probleemloos kunnen interfaceren met een waslijst aan andere systemen (barcodelezers, weegschalen, betaalterminals, de systemen van stockbeheer, productinformatie, finance, customer relations, marketing & promotie etc.). Een robuust testsysteem is

bijgevolg onontbeerlijk en ondersteunt de softwareontwikkeling door:

- het vroegtijdig onderscheppen van defecten
- een grotere test coverage
- een grotere efficiëntie van testen
- een snellere time-to-market
- een verlaging van de ontwikkelingskosten

De eerste stap voor het uitbouwen van een testsysteem is het kiezen van een gepast testing framework. Ook voor een relatief jong framework zoals Angular zijn er reeds verschillende alternatieven beschikbaar. Het onderzoek dat

onderwerp is van dit voorstel bestaat er dus uit om na te gaan:

- welke e2e (end to end, d.w.z. testen die gebruik in een productie-omgeving simuleren) testing frameworks bestaan voor Angular en welke randvoorwaarden (kost, beschikbare ondersteuning en documentatie, vereiste programmeervaardigheden) zij hebben,
- in welke mate de bestaande frameworks de volledige functionaliteit van het systeem kunnen afdekken,
- hoe performant elk van de frameworks is en
- hoe betrouwbaar de resultaten van de frameworks zijn

om zo het meest geschikte framework voor de use case van Colruyt Group te kunnen aanbevelen.

## 2. State-of-the-art

De keuze voor een geschikt e2e testing framework is sterk afhankelijk van de specifieke context waarbinnen dit ge-

bruikt gaat worden. Colruyt Group overweegt momenteel 2 verschillende frameworks:

- **Protractor** — een API voor e2e testing die steunt op Selenium (een “WebDriver” die webapplicaties voor testdoeleinden automatiseert)
- **Cypress** — een relatief nieuw e2e framework

Daarnaast zijn er nog een reeks andere Behavior-Driven Development of testing frameworks voor het JavaScript ecosysteem. De meest prominente frameworks die van toepassing zijn op Angular, buiten bovenstaande, zijn de volgende:

- **Jasmine**
- **Jest**
- **Mocha**
- **Puppeteer**
- **AVA**
- **Cucumber**

Zie ook Tulio Castro, 2019; Definition, 2019; Ali Kamalizade, 2019; Nwose Lotanna, 2019; Arnab Roy, 2019 en Vitali Zaidman, 2019.

### 3. Methodologie

Om het meest geschikte framework voor Colruyt Group's checkoutsysteem te kunnen identificeren, dienen volgende stappen genomen te worden:

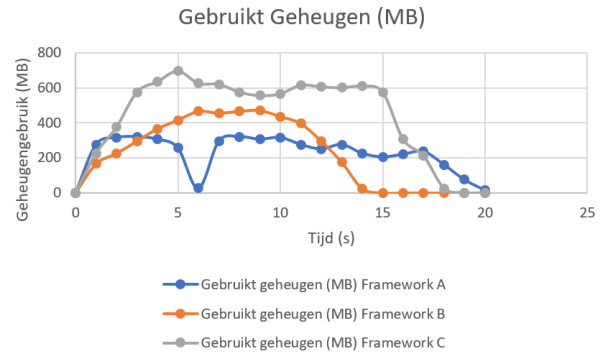
1. onderzoek naar de **mogelijkheden en randvoorwaarden** van elke kandidaat, om daar de 3 meest geschikte kandidaten uit te halen
2. onderzoek naar de **huidige functionaliteit** van het kassasysteem, alsook de bestaande interfacing met randapparaten en andere systemen
3. de selectie van een **representatieve subset** van de functionaliteit van het kassasysteem
4. schrijven van **identieke testscenario's** in elk van de 3 gekozen testing frameworks
5. opzetten van de frameworks, **uitvoeren van de testscenario's** en meten van de performantie (complexiteit van installatie, benodigde tijd om testen te schrijven, doorlooptijd, geheugengebruik)
6. formuleren van een **advies** op basis van de gegevens verzameld in de voorgaande stappen

Dit advies is het eindresultaat van dit onderzoek.

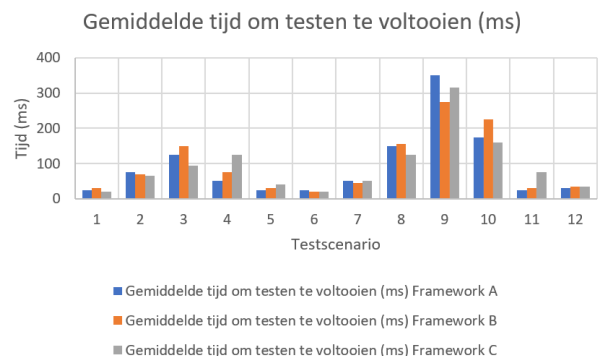
### 4. Verwachte resultaten

De resultaten voor dit onderzoek zijn moeilijk te anticiperen omwille van het veelvoud aan variabelen die van toepassing zijn. Elk van de verschillende testingframeworks zal vermoedelijk voor- en nadelen hebben en het uiteindelijke advies zal afhangen van de specifieke behoeften van Colruyt Group en de eigenschappen van het Checkoutsysteem.

Naast een overzicht van de mogelijkheden en randvoorwaarden van elk testing framework, zullen de resultaten van de testscenario's een objectief beeld van de voor- en nadelen van elk framework kunnen geven.



**Figuur 1.** Mock grafiek: Geheugengebruik in MB van de verschillende testframeworks tijdens het uitvoeren van de testscenario's.



**Figuur 2.** Mock grafiek: Benodigde tijd om elk van de verschillende testscenario's uit te voeren.

### 5. Verwachte conclusies

Zoals reeds eerder vermeldt, is het moeilijk om in te schatten welk framework uiteindelijk geadviseerd zal worden. De ontwikkelaars van het Checkoutsysteem lijkt in eerste instantie de voorkeur te geven aan het recentere Cypress, vooral omdat testen schrijven in dit framework intuïtiever en sneller zou zijn. In welke mate een framework aanvaard wordt is vaak de doorslaggevende factor om een bepaald framework te kiezen, en het lijkt dus aannemelijk dat Cypress uiteindelijk opgenomen wordt in het advies.

### Referenties

- Ali Kamalizade. (2019). How to use Jest in Angular aka make unit testing great (again). <http://definitiontech.co/top-4-unit-testing-tools-for-angular-and-react-applications/>. [Online; accessed 14-November-2019].
- Arnab Roy. (2019). Top Eight JavaScript Automation Testing Frameworks In 2019. <https://dzone.com/articles/top-8-javascript-automation-testing-frameworks-in/>. [Online; accessed 14-November-2019].
- Definition. (2019). Top 4 Unit Testing Tools for Angular and React Applications. <http://definitiontech.co/top-4-unit-testing-tools-for-angular-and-react-applications/>. [Online; accessed 14-November-2019].
- Nwose Lotanna. (2019). Top Javascript Testing Frameworks in Demand for 2019. <https://blog.bitsrc.io/top->

- javascript-testing-frameworks-in-demand-for-2019-90c76e7777e9. [Online; accessed 14-November-2019].  
softwaretestinghelp.com. (2019). What Is End To End Testing: E2E Testing Framework With Examples. <https://www.softwaretestinghelp.com/what-is-end-to-end-testing/>. [Online; accessed 8-December-2019].  
Tulio Castro. (2019). End to End (e2e) – Angular Testing – Protractor vs Cypress. <https://techblog.fexcofts.com/2018/09/24/end-to-end-e2e-angular-testing-protractor-vs-cypress/>. [Online; accessed 15-November-2019].  
Vitali Zaidman. (2019). An Overview of JavaScript Testing in 2019. <https://medium.com/welldone-software/an-overview-of-javascript-testing-in-2019-264e19514d0a>. [Online; accessed 14-November-2019].