

Co-reflection & Co-creation

Introductie

In dit onderzoeksdocument vind je de resultaten en conclusie over hoe BBD het beste de front-end componenten library kan gebruiken.

Introductie	1
Doel	2
Motivatie	2
Onderzoeksvraag	2
Hypothese	2
Deelvragen	2
Plan van aanpak	3
Testplan	3
Testgroep	3
Ingrediënten/voorbereiding	3
Meeting indeling	3
Opdracht	4
Uitnodiging	4
Resultaten	4
Conclusie & Aanbevelingen	5

Doel

Het hoofddoel is om erachter te komen hoe de front-end componenten library op een efficiënte manier ingezet kan worden zodat de front-end developers er op de lange termijn voordeel uit kunnen halen. Een subdoel is om de front-end developers kennis te laten maken met de mogelijke oplossing zodat het in de toekomst duidelijker is hoe de componenten library werkt. Hierbij wordt ook gekeken of de developers de documentatie goed kunnen volgen.

Motivatie

Als bekend is hoe de front-end developers het zelf graag zien, kan de componenten library zo ontwikkeld worden dat de developers gemotiveerd zijn deze te onderhouden. De front-end developers zullen niet precies weten wat ze willen voordat ze een keer gebruikgemaakt hebben van de componenten library. Daarom is het van belang dit onderzoek uit te voeren in de vorm van testen.

Onderzoeksvraag

Welke mogelijkheden zien de BBD front-end developers bij het gebruiken van de front-end componenten library?

Hypothese

Er wordt verwacht dat de BBD front-end developers moeite hebben met het ontwikkelen van herbruikbare componenten aangezien zij in de huidige situatie voornamelijk project specifieke componenten ontwikkelen. Er wordt verwacht dat de front-end developers guidelines nodig hebben om herbruikbare componenten te kunnen ontwikkelen die één bepaalde structuur aanhouden. Verder wordt er verwacht dat de BBD front-end developers uit kunnen leggen wat voor componenten ze willen hergebruiken (Shopware of Magento, molecules, atoms, organisms, etc.).

Deelvragen

- Tegen welke problemen lopen de front-end developers aan tijdens het uitvoeren van de opdracht? Tegen welke problemen lopen de front-end developers aan wanneer zij de documentatie volgen?
- Welke tips/feedback geven de front-end developers om het proces/het gebruiksgemak te verbeteren?
- Welke front-end componenten vinden de front-end developers nuttig om te kunnen hergebruiken? Welke front-end componenten vinden de front-end developers niet nuttig genoeg om te hergebruiken?
- Vinden de front-end developers het ontwikkelen en hergebruiken van Magento componenten waardevoller dan het gebruiken van Shopware componenten?
- Hoe complex/uitgebreid vinden de front-end developers dat de componenten moeten zijn (met betrekking tot Atomic design)? Wat is de 'sweet spot'?

Plan van aanpak

Showroom en workshop research wordt toegepast.

Het plan is om samen met de front-end developers van BBD een aantal uur in een ruimte te zitten om te brainstormen hoe de front-end componenten gebruikt moet worden. Dit wordt bereikt door een korte uitleg d.m.v. een Use Case demo te geven om vervolgens te discussiëren wat de front-end developers hiervan vinden. Op basis van de feedback gaan de front-end developers componenten ontwikkelen op de manier die zij het beste lijken en deze uploaden naar de library met de bijbehorende documentatie.

Deelnemers

De deelnemers moeten voldoen aan de volgende eisen:

- Bluebird Day front-end developer zijn
- De front-end developers moeten uit minsten twee verschillende teams komen, het liefste drie verschillende teams.

Ingrediënten/voorbereiding

- Een meeting van 2/2.5 uur in een gesloten ruimte inplannen
- 3 of 4 front-end developers uit verschillende teams.
- Een uitnodiging versturen naar de front-end developers met daarin uitleg over de meeting (bijv. het doel)

Meeting indeling

1. Introductie: herhaal wat het doel is van de meeting en wat we gaan doen om dat doel te bereiken.
2. Uitleg over de oplossing/front-end componenten library
3. Vragen beantwoorden.
4. Vragen en discussiëren met de developers over hoe complex de componenten moeten zijn.
5. De developers vragen om ieder een veelgebruikt/nuttig component te maken en te uploaden naar de library.
6. Vragen hoe de developers de opdracht ervaren hebben. Waar liepen de developers tegenaan?

Opdracht

1. Maak/gebruik een veelgebruikt/nuttig component
2. Upload de component naar de library
3. Voeg de nodige documentatie toe. Zorg voor:
 - a. Een uitleg over waar het component voor gebruikt wordt
 - b. API info
 - c. Eventueel best practices

Uitnodiging

Hallo beste collega's,

Zoals de meesten van jullie weten ben ik de afgelopen maanden actief bezig geweest met het ontwikkelen van een front-end componenten library. Op dit moment staat er een front-end componenten library omgeving klaar, maar deze is (op de installatie documentatie na) nog helemaal leeg.

Daar hebben jullie natuurlijk niets aan.

Daarom lijkt het mij goed om een meeting te houden waarin we met een kleine groep front-end developers gaan bepalen wat voor componenten jullie handig vinden om te kunnen hergebruiken in toekomstige projecten. Tijdens deze meeting wordt dan ook meteen duidelijk hoe complex de componenten zullen zijn.

Het liefste zien we één developer van Team 1, 2 en 3 hun team vertegenwoordigen.

Vul a.u.b. de datumprikker in als je wilt helpen.

Resultaten

Tegen welke problemen lopen de front-end developers aan tijdens het uitvoeren van de opdracht? Tegen welke problemen lopen de front-end developers aan wanneer zij de documentatie volgen?

- De developers kregen soms (node) dependency errors. Dit had te maken met de verschillende versies.
- Het lukte de developers niet om hun README terug te kunnen zien in de navigatie van het documenten platform. Dit komt omdat er instructies misten in de documentatie.

Welke tips/feedback geven de front-end developers om het proces/het gebruiksgemak te verbeteren?

- Probeer een design systeem voor het managen van componenten toe te passen. Het [Carbon Design System](#) bijvoorbeeld.
- Voor het managen van nested node dependencies, kun je onderzoek doen naar [NPM workspaces](#)

Welke front-end componenten vinden de front-end developers nuttig om te kunnen hergebruiken? Welke front-end componenten vinden de front-end developers niet nuttig genoeg om te hergebruiken?

De volgende front-end componenten zijn regelmatig nodig in de projecten van Bluebird Day en zijn dus gewenst:

- Tabs-accordion
- Typografie
- Icons
- Kleur variabelen
- Image
- Buttons
- Modal
- Menu's
- Form
- Containers/layouts
- Breadcrumbs
- Card
- Slider/carousel
- Producttegel
- Tooltips
- Pagination
- Filtering - search bars/checkboxes
- Product labels
- Messages
- Drawer
- Animaties - mixins/base classes
- Footer
- Header
- Banner
- Video

Daarnaast verwachten de front-end developers ook dat er componenten zijn die zorgen voor een duidelijkere structuur in projecten. De manier waarop scripts worden ingeladen kan bijvoorbeeld ook in de vorm van een code snippet getoond worden op het platform.

Front-end Standaarden

- Scripts inladen
- Lint

Vinden de front-end developers het ontwikkelen en hergebruiken van Magento componenten waardevoller dan het gebruiken van Shopware componenten?

De front-end developers hebben aangegeven dat ze voor beide ecommerce platformen regelmatig custom componenten moeten maken. Het enige verschil is dat dit met Magento vaak meer complex is dan met Shopware. De front-end developers wensen componenten te kunnen hergebruiken voor beide platformen en ook voor Vue Storefront in de toekomst.

Hoe complex/uitgebreid vinden de front-end developers dat de componenten moeten zijn (met betrekking tot Atomic design)? Wat is de 'sweet spot'?

Tijdens de meeting is hier niet tot in de details over gepraat. Wat de developers vooral handig vinden zijn atoms en molecules. Organisms zoals footers en headers zijn ook welkom. Dit blijkt uit de lijst van componenten die tijdens de meeting opgesteld werd.

Conclusie & Aanbevelingen

Nu het onderzoek klaar is, kan de hoofdvraag worden beantwoordt.

“Welke mogelijkheden zien de BBD front-end developers bij het gebruiken van de front-end componenten library?”

De developers vinden de basis van de oplossing goed. Echter zien zij mogelijkheden om de oplossing te verbeteren door een design system toe te passen op de omgeving om te zorgen voor een betere structuur.

Ook voor het managen van versies is een oplossing nodig om versie problemen tussen de componenten te voorkomen.

Tijdens de meeting bleek ook dat de developers in de huidige situatie niet geheel tevreden zijn over het gebruik van Confluence vanwege het gebrek aan technische documentatie support. Er kan dus ook geconcludeerd worden dat Confluence geen geschikte plaats is voor de documentatie van de componenten library. Tijdens mijn onderzoek naar MkDocs, ondervond ik dat MkDocs ook niet ideaal is om te gebruiken als documentatie platform.

Wat nu?

Er moet onderzoek verricht worden naar geschikte design systemen voor de oplossing.

Daarnaast moet onderzocht worden welke documentatie platform hierbij past.

Verder moet er onderzocht worden hoe de versies van nested components gemanaged kunnen worden.