



- **Introductie 13.45 – 14.30**
- **Aan het werk 😊 14.30 – 16.30**
- **Conclusie / restrospective 16.30 – 17.00**



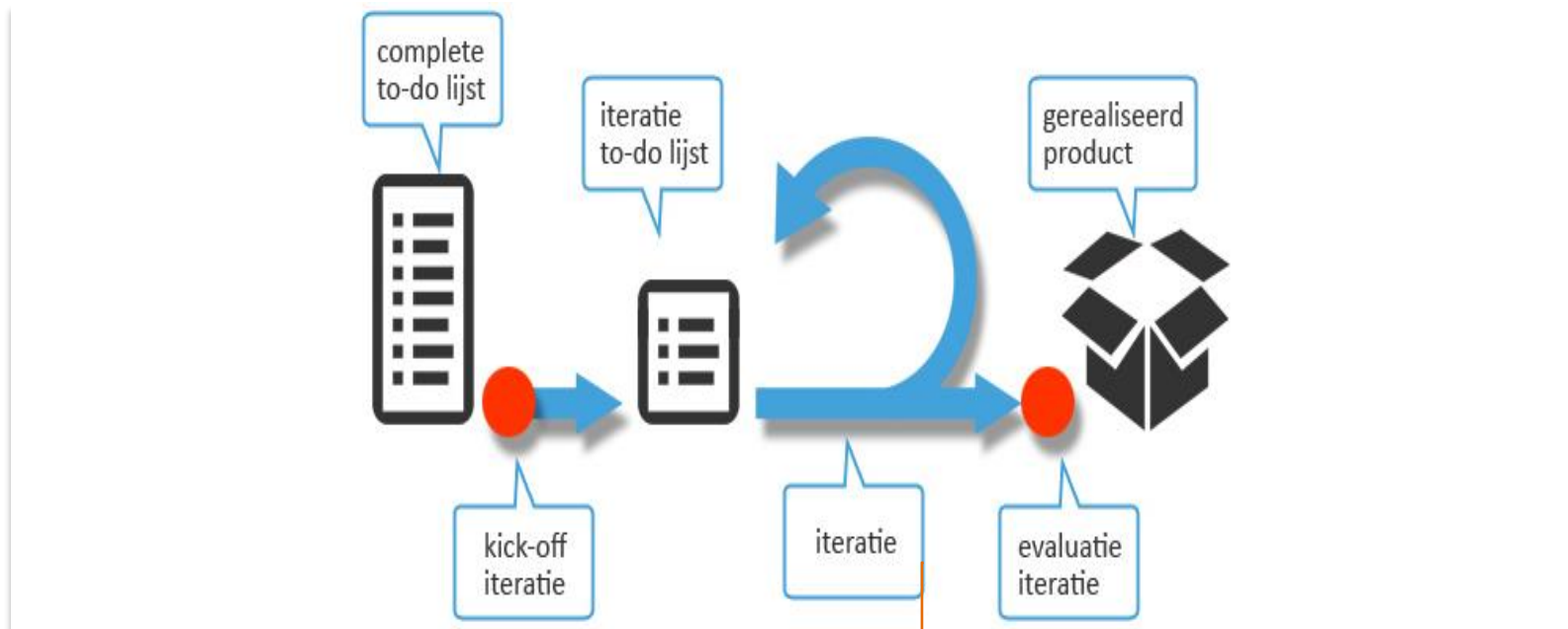
- Voorstellen

- Waar ben je echt goed in (3 minuten)

- Teams vormen op basis van het voorstellen (2 minuten)

- Iteratief werken





- Elke iteratie/sprint duurt 30 minuten
- Een iteratie/sprint bevat alle noodzakelijke taken: planning, analyse, ontwerp, testen en documentatie.
- Aan het eind van de iteratie wordt het product getoond, getest en beoordeeld
- Teams visualiseren wat ze aan het doen zijn.



In agile projecten worden Features en User Stories gebruikt om de requirements vast te leggen. Idealiter bevat elke US acceptatiecriteria die worden gebruikt als basis voor het ontwikkelen van testgevallen. Deze sectie beschrijft de aanpak voorbereiden en uitvoeren van test cases

Stap	Korte beschrijving	Benodigde rollen
Discussieer	Refinement sessies om een gedeeld begrip van de benodigde Feature/ user stories te krijgen. Identificeer real-life scenario's en voorbeelden met een realistische context	<ul style="list-style-type: none"> • Klant • Business Analyst • Developer • Tester • Product Owner
Specificeer en leg vast	<p>Leg de benodigde requirements vast in een uitvoerbaar specificatie op basis van (en binnen de user stories) voorbeelden en acceptatiecriteria. Specificaties zijn vastgelegd in het volgende voor iedereen te begrijpen formaat:</p> <p>Feature/User Story: Titel In order to [value] As a [role] I want [feature]</p> <p>Scenario Outline: Titel Given [context] And [more context] When [event] Then [outcome] And [another outcome]</p>	<ul style="list-style-type: none"> • Analist • Developer • Tester • (Klant)
Ontwikkel	Ontwikkel de gewenste Features/User Stories met behulp van geautomatiseerde Unit-test scenario's. Geautomatiseerde acceptatietests worden gebouwd rond de geïdentificeerde scenario's. Geautomatiseerde unit en integratie testen worden gebruikt om de uitvoering te ondersteunen	<ul style="list-style-type: none"> • Developer • Tester
Demonstreer	Demonstreer de implementatie/oplevering door het uitvoeren van de acceptatietesten en het uitvoeren van handmatige Exploratory testen	<ul style="list-style-type: none"> • Developer • (Klant) • Product Owner

Onze principes

- Iedereen is verantwoordelijk voor test en kwaliteit
- Alle type testen (Unit, Integratie, Acceptatie, regressie, performance, etc.) worden geautomatiseerd tenzij anders aangegeven. Handmatig testen wordt alleen gebruikt bij exploratory testen tenzij anders aangegeven
- Test cases, Code, Documenten en data worden met dezelfde importantie als productie code behandeld
- Implementeer als eerst de hoogst geprioriteerde Werk Items. (Elk nieuw werk item is geprioriteerd door de PO/Team Manager en aan de Backlog toegevoegd)
- Werk Items mogen op elk moment opnieuw worden geprioriteerd, of verwijderd
- Een werk item met meer detail heeft een grotere prioriteit, dan een module met minder detail

Test Scope

In scope

- Unit Testen
- Model testen
- Component testen
- Functionele test
- API test
- Performance testen
- Keten testen
- Regressie testen
- Usability testen
- Exploratory testen
- Usability testen

Out of scope

- Third Party Systems

Kwaliteits attributen en Risico klasse

Risicoklasse 2 Foutbestendigheid Continuïteit Inpasbaarheid	Risicoklasse 3 Onderhoudbaarheid	Risicoklasse 3 Functionele juistheid
Risicoklasse 1	Risicoklasse 2 Performance Koppelbaarheid Usability	Risicoklasse 3 Herstelbaarheid
Risicoklasse 1 Functionele volledigheid Security	Risicoklasse 1	Risicoklasse 2 Overdraagbaarheid Flexibiliteit

Acceptatie bij

100% gereedheid van afgesproken features /US met open
Severity 1 defects = 0
Severity 2 defects = 5
Severity 3 defects < 5
Severity 4 defects < 10

De risicoklassen

Voor elk risico uit de risicoanalyse en risico matrix is de risicoklasse sturend voor het bepalen van de inspanning van de testen

- Risicoklasse 1 - laag risico
- Risicoklasse 2 - middel risico
- Risicoklasse 3 – hoog risico
- Risicoklasse 0 - Technisch

Testtools

Testware

- Agile test aanpak
- Reporting
- TER

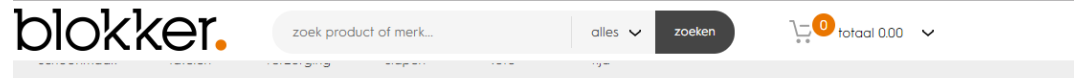
Defect Management

- TFS/Planon

- Unit Testen
 - TFS/Planon
- Model testen
 - TFS/Planon
- Component testen
 - TFS/Planon
- Functionele test
 - TFS/Planon
- API test
 - TFS/Planon
- Performance testen
 - TFS/Planon
- Keten testen
 - TFS/Planon
- Regressie testen
 - TFS/Planon
- Usability testen
 - TFS/Planon

- **Introductie 13.45 – 14.30**
- **Aan het werk 😊 14.30 – 16.30**
- **Conclusie / restrospective 16.30 – 17.00**





op werkdagen voor 22.00u besteld, volgende dag in huis* gratis bezorgd in één van onze winkels* gratis thuisbezorgd vanaf €25* gratis retour: 30 dagen bedenktijd*

brandpreventiemaand **is jouw huis al veilig?**  [bekijk nu >](#)

@brabantia
nieuwe kleuren
nieuwe stijl.
bekijk hier alle Touch Bins >



web
only

oktober
brandpreventiemaand

ontdek de nieuwe
touch bin's van brabantia

tefal wisselweken
20% korting

koffiemachines
die bij je passen

ONTDEK MIJN BLOKKER



Versie : High Performance Testing

Datum : 4 juli 2018

Door : Nick & Jochem

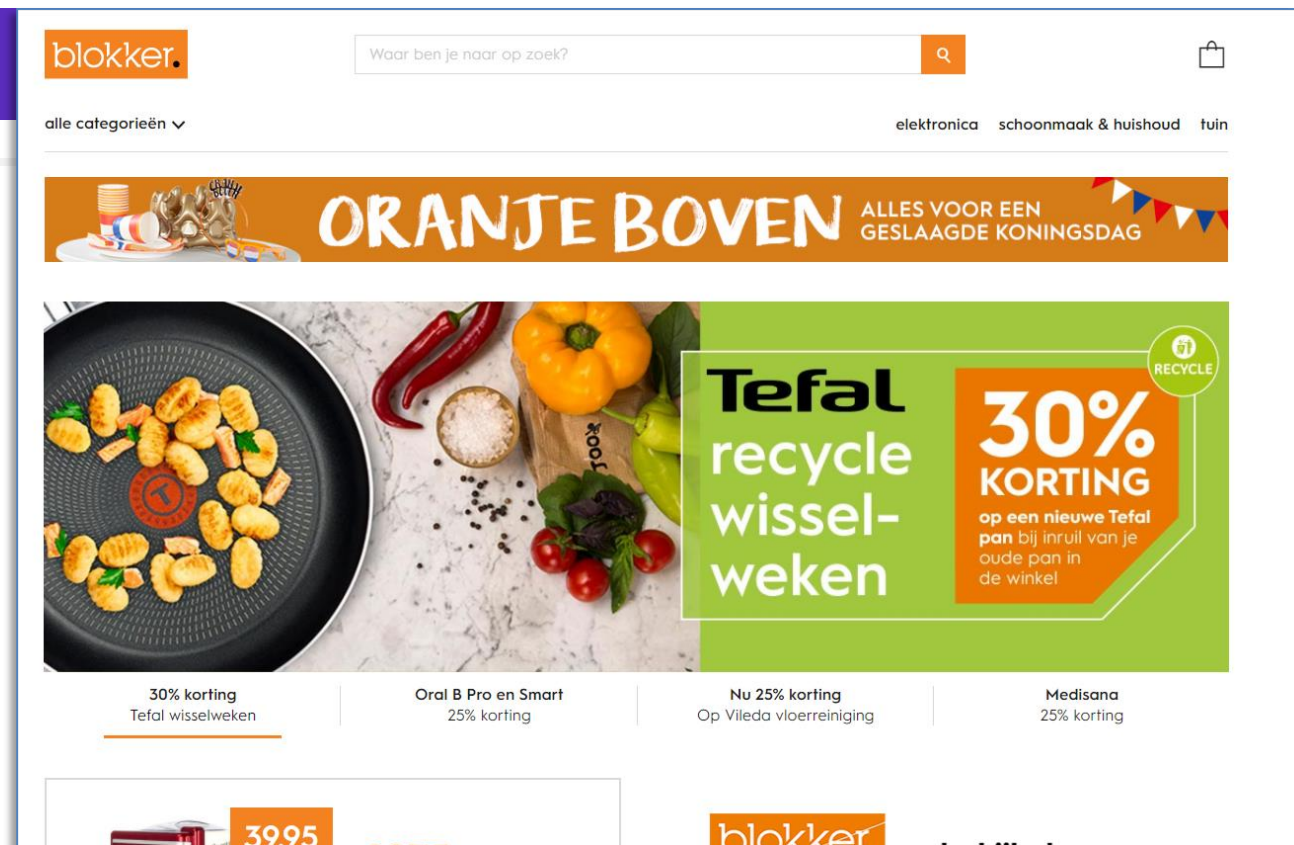




BLOKKER HOLDING



- Versterken van het merk Blokker
- Aanzetten tot winkelbezoek
- Directe omzet + marge



De site is de buitenste schil van een landschap. Sinds maart 2018 totaal vernieuwde technologie, onder meer om continuous development mogelijk te maken.

Deze nieuwe website is het onderwerp van vandaag.

Soft evaluation

1. Interactie: hoe werkt het?
2. Zoekresultaat: correct en volledig?
3. Bestelproces: is dat eenduidig?

Hard evaluation

1. Zijn alle producten via de menu structuur te vinden?
2. Zijn alle bestelbare producten daadwerkelijk op voorraad?
3. Welke pagina's hebben een page load > 5 sec.? Onder welke condities?



Test strategy

Tijdslot: 3 x 25 minuten (5 minuten voor de op te leveren producten)



Bug report

Eenmaal per kwartier is het mogelijk te spreken met de Product Owner (Nick & Jochem)



Test advice

Veldnaam	Invulling
Issue Type	Bug
Summary	Naamgeving van de betreffende bevinding
Reporter	Indiener van de bevinding
Description	<ol style="list-style-type: none"> Probleem korte omschrijving van het probleem. Verwachting Omschrijving van wat ik van de functionaliteit verwacht had, waarin duidelijk het verschil met het PROBLEEM wordt aangekaart. Reproductiepad Een duidelijk pad, met daarin o.a. (indien mogelijk) verwerkt op welke omgeving met welke rollen in welk (deel) van het proces wat gebeurt. Doel is om ervoor te zorgen dat iedereen die de bevinding leest zonder voorkennis dit pad kan volgen (mits de juiste autorisaties etc.). Denk hierbij aan productnummer, het gevolgde testscript (toevoegen!) evt. schermprintjes, etc. De ernst van de bevinding (t.b.v .prio): Omschrijving van wie hierdoor worden geraakt, wat er mis gaat als het niet wordt opgelost, of er een (tijdelijke) workaround mogelijk is.
Fix version	<ol style="list-style-type: none"> Het verantwoordelijke team/Afdeling De betreffende Release (Fase) waarin de bevindingstory thuishoort (indien bekend).

Belangrijke uitgangspunten

- Een bevinding wordt geregistreerd als issuetype 'BUG' (en dus niet als userstory)
 - Op deze manier is filteren op 'Bugs' mogelijk
 - Dit helpt om inzicht te krijgen in het aantal bevindingen per bv fase
- Een bevinding is altijd gekoppeld zijn aan een userstory
- **Let op:** Bevindingen worden omgezet naar stories indien ze meegenomen worden naar een volgende fase en dus niet worden opgepakt in de fase waarin ze tijdens testwerk gevonden zijn.
- De bevinding wordt administratief gesloten en "gecloned" naar een nieuwe story.

Veldnaam	Invulling
Components	Zoekmachine
Assignee	Naam medewerker verantwoordelijk
Issue	Koppeling van story waarop de bevinding gevonden is

- **Introductie 13.45 – 14.30**
- **Aan het werk 😊 14.30 – 16.30**
- **Conclusie / restrospective 16.30 – 17.00**



- 5 minuten pp; – top, -tip, -flop
- bundelen en bespreken
- Welk advies zouden jullie blokker geven
- -hoog, - middel, -laag. –technische bevindingen

