

# TEST STRATEGIE

## TEAM APP

<b>Project naam</b>	112 NL – Team app (DPC)
<b>Datum</b>	27-04-2020
<b>Project Manager</b>	10.2.e

<b>Versie nummer</b>	1.1
<b>Draft/Final as of</b>	
<b>Auteur</b>	10.2.e 10.2.e

Goedkeuring Handtekeningen		
Naam	Titel	Bevestiging
10.2.e	10.2.e	
10.2.e	10.2.e )	
10.2.e	10.2.e	
10.2.e	10.2.e	Ja
10.2.e	10.2.e	Pending

# 112 NL

Test plan – Team App (DPC)

Document control		
Naam	Titel	Distribution
10.2.e	10.2.e	Goedkeuring
10.2.e	10.2.e )	Goedkeuring
10.2.e	10.2.e	Goedkeuring
10.2.e	10.2.e	Goedkeuring
10.2.e	10.2.e	Review
10.2.e	10.2.e	Review
10.2.e	10.2.e	Review

Revision history				
Datum	Versie	Auteur	Gereviewed door	Aangepast
21-4-2020	V1	10.2.e	10.2.e	Comments inserted
7-5-2020	V1	10.2.e	10.2.e	Comments inserted
2-6-2020	V1	10.2.e	10.2.e	Visualisation test scope
11-6-2020	V1	10.2.e	10.2.e	Performance

## INHOUDESOPGAVE

1	INTRODUCTIE.....	4
1.1	Aannames & voorwaarden .....	4
2	SYSTEM OVERVIEW .....	5
3	SCOPE .....	6
3.1	Third parties.....	9
3.2	Out of scope.....	9
4	TEST PLAN .....	10
4.1	Test levels .....	10
4.2	Regressie test.....	10
4.3	Productie Acceptatie Test.....	11
4.4	Definition of Ready / Definition of Done .....	11
4.5	Test tools .....	12
5	TEST BENADERING .....	14
5.1	Test Deliverables .....	14
5.2	Test devices.....	15
5.3	Test data .....	16
6	TEST OMGEVINGEN.....	17
7	TESTMANAGEMENT & RAPPORTEREN .....	18
7.1	Test rapport .....	18
7.2	Bugs en bevindingen management .....	18
7.3	Defect classificatie .....	18
7.4	Story bug flow .....	19
7.5	Bug flow.....	19
7.6	Risico's.....	19
8	TEST ACTIVITEITEN EN SCHEMA .....	21
9	ROLLEN EN VERANTWOORDELIJKHEDEN.....	22
10	APPENDIX.....	23

## 1 INTRODUCTIE

Het doel van dit document is om te definiëren hoe Team App haar scope in het 112 NL project gaat testen om de kwaliteit te waarborgen. In dit testplan worden de testdoelen en taken beschreven.

In dit testplan wordt uiteengezet:

- De verschillende fases van testen
- Welke methoden van testen worden toegepast

Dit document is opgesteld om (Project) Managers, Testers en Developers te informeren over het test proces. Een testplan legt uit hoe testen in zijn algemeenheid, op specifieke features en voor go-lives worden uitgevoerd. Dit testplan geldt alleen voor Team app van DPC. Het testplan van 112 integrale ketentest en User Acceptance Test wordt uitgevoerd onder leiding van [10.2 e](#).

### 1.1 Aannames & voorwaarden

De volgende aannames en voorwaarden zijn van toepassing voor het testen van het 112 NL project:

- De personen die dit document dienen te beoordelen zijn onderdeel van LMS, de Politie of Team app.
- Het testplan wordt voor start testwerkzaamheden goedgekeurd.
- De Product Owner & Business Owner vanuit LMS en de Politie dragen zorg voor ondersteuning vanuit beide organisaties met betrekking tot vereisten, sign-off-activiteiten, prioritering van taken en problemen.
- Test omgevingen voor alle interfaces en technische koppelingen die onderdeel zijn van dit project, moeten op tijd beschikbaar zijn om de testen op uit te voeren.
- Business vertegenwoordigers zullen feedback geven over de opgeleverde onderdelen.
- JIRA en Confluence zijn gedurende de gehele testfase beschikbaar. Dit zijn de primaire kanalen voor de opslag en documentatie van de testresultaten.

## 2 SYSTEM OVERVIEW

De 'System under Test (SUT)' bestaat voor Team App uit de volgende scope (groene onderdelen):\*

10.2.g



\*Dit is nog niet de final SUT. Mocht er een update van de plaat ontstaan, dan zal er een nieuwe versie van het testplan komen.

## 3 SCOPE

In de volgende visualisering is te zien welke onderdelen van de scope door team Team App developed en getest zullen worden (**groen**).

10.2.g



Elementen die tijdens het testen wel meegenomen worden, zoals API Gateway, worden wel getest maar het oplossen van eventuele defecten valt onder desbetreffende partij.

De ketentest – van mobile device tot Meldkamer – valt buiten de primaire scope van Team App. Uiteraard werkt Team App mee met het projectteam voor de inrichting hiervan.

# 112 NL

Test plan – Team App (DPC)

Binnen de scope van de 112NL native app, vallen de volgende functionaliteiten:

10.2.g



# 112 NL

Test plan – Team App (DPC)

10.2.g





## 3.1 Third parties

De integratie en de compatibiliteit van de derde partijen zal worden getest. Het (eventueel) oplossen van bevindingen of opleveren van functionaliteiten van de third party applicaties zijn niet in scope van Team App. Bevindingen die door Team App worden gevonden bij een derde partij, zal door de desbetreffende partij worden opgelost. Vanuit Team app is deze afspraak onderdeel van de Acceptatiecriteria.

Third party partijen en systemen binnen het 112NL project zijn:

- 10.2.g (IT infra inrichting, hosting en beheer) van het Politie WAH platform
- LMS voor de GMS Webservice en infra
- Meldkamer voor het GMS systeem en infra
- Google/ Apple voor ALM HTTPS functionaliteit en het aanpassen van de native OS versies.
- KPN Routing voor het bellen naar de meldkamer
- 10.2.g als SMS broker voor AML SMS

## 3.2 Out of scope

De grijze onderdelen in de System Overview (hoofdstuk 2) zijn out of scope voor development en testing.

**De betekent dat volgende test types out of scope zijn voor Team App:**

- **Security – met betrekking tot de gehele keten**
- **Performance (loadtest, stresstest, volume test) - met betrekking tot de gehele keten**
- **UAT – validatie van oplossing door gebruikers**
- **Functionele ketentest - app tot meldkamer**

Devices, zoals tablets, en OS versies die niet in scope zijn opgenomen zullen niet worden getest.

Functionaliteiten van derden die niet in scope zijn van de integration of system testing vallen ook buiten de scope. Onderstaand lijst toont de verhoudingen binnen het project en bijbehorend QA.

- Team App & Services – 10.2.e
- 10.2.g – 10.2.e
- MDC 112 – 10.2.e
- GMS webservice – 10.2.e
- 112 Keten – 10.2.e

## 4 TEST PLAN

Dit hoofdstuk beschrijft de test aanpak voor het 112 NL project. Het 112 NL project is een Continuous Integration/ Continuous Development project. Enerzijds omvat het test plan de aanpak tijdens de sprints en als onderdeel van de CI/CD. Anderzijds omvat het testplan de benodigde tests die voor livegang voor beide apps en services uitgevoerd zullen worden. Het functioneren van de services is afhankelijk van de rest van de keten. Livegang van de keten wordt omschreven in het generieke 112 Testplan.

### 4.1 Test levels

Test Area	Test level	Environment	When	Main Stakeholder
<b>Development team</b>	Unit testing	DEV	Continuous/ elke sprint	DPC Team app Developers
	Integration testing	DEV	Continuous/ elke sprint	DPC Team App - Developers
	System testing	QA	Na oplevering op Acc	DPC Team App - Tester
	UI testing	QA	Continuous/ elke sprint	DPC Team App - Tester
	Performance test (mbt de app & servers)	ACC	Op Dev en Acc omgeving	DPC Team App Developers & tester
	Regression testing	ACC	Na oplevering op Acc	DPC Team App - Tester
	Smoke testing	PROD	Na oplevering op Prod	DPC Team App - Tester
	Bug hunt	ACC	Na oplevering op Acc	DPC Team App
	Bèta test	ACC	Na oplevering op Acc	DPC Team App
<b>Business User test</b>	User Acceptance Test	ACC	Na oplevering op Acc	LMS
<b>Security test</b>	User Acceptance Test	PROD	Na oplevering op Acc	Externe partij onder aansturing van DPC System Team
<b>Performance test</b>	User Acceptance test	PROD	Na oplevering op Acc	LMS/ externe partij

### 4.2 Regressie test

Team App zal voor haar scope een end-to-end (van app tot services) regressietest opstellen. De regressie testset bevat standaard de kritieke tests van eerder gereleasde functies. De manuele regressie testset staat in Confluence. De automatische regressie testset is opgeslagen in de tool Eclipse x Robotframework, die geconfigureerd is met Gitlab.

De regressie test bestaat uit 2 niveaus:

- Smoketest – een selectie van tests zullen uitgevoerd worden om te verzekeren dat het basis systeem voldoet aan de requirements en dat er geen onverwachtse bugs zullen optreden. Smoketests wordt uitgevoerd als onderdeel van het hertesten van bugfixes.
- Regressietest – deze test zal worden uitgevoerd als onderdeel van de systemtest op de test omgeving. Ook wordt de regressietest ingezet als acceptatie test voor deployment naar Productie omgeving. Het doel van de regressietest is om te testen of later gereleasde code impact heeft gehad code die eerder is geschreven en getest.

Er zal een risico analyse plaatsvinden welke onderdelen in de regressie testset worden opgenomen.

### 4.3 Productie Acceptatie Test

Nadat de Regressie test op acceptatie is voltooid kan de code ingediend worden bij de Google Playstore en App Store. Onderdeel van de Productie Acceptatie Test is:

- Regressietest op productie
- Voldoen aan Security requirements voor app en services
- Voldoen aan Performance requirements voor app en services
- Voldoen aan Business requirements voor app en services

### 4.4 Definition of Ready / Definition of Done

Test Area	Test level	DOR	DOD
<b>Development team</b>	Unit testing	<ul style="list-style-type: none"> <li>· Testomgeving beschikbaar</li> <li>· Stories in de sprintbacklog</li> </ul>	<ul style="list-style-type: none"> <li>· Alle unit tests zijn geslaagd</li> <li>· 80% code coverage</li> <li>· Sonarcube triple A status</li> </ul>
	Integration testing	<ul style="list-style-type: none"> <li>· Testomgeving beschikbaar</li> </ul>	<ul style="list-style-type: none"> <li>· Elke backend service is getest als 1 groep van componenten</li> </ul>
	System testing	<ul style="list-style-type: none"> <li>· Testdevices beschikbaar op vpn</li> <li>· Buildlaptop beschikbaar</li> </ul>	<ul style="list-style-type: none"> <li>· Automated system test is geslaagd</li> </ul>
	UI testing	<ul style="list-style-type: none"> <li>· Testomgeving bereikbaar</li> <li>· Stubs beschikbaar</li> <li>· Testdevices beschikbaar</li> <li>· Status Jira: Available for testing</li> </ul>	<ul style="list-style-type: none"> <li>· Voldaan aan acceptatie criteria in de story</li> </ul>
	Performance test	<ul style="list-style-type: none"> <li>· Requirements zijn bekend</li> </ul>	<ul style="list-style-type: none"> <li>· Stress-, load- en volume test zijn geslaagd</li> </ul>
	Regressie test	<ul style="list-style-type: none"> <li>· Testdevices beschikbaar</li> <li>· Business requirements</li> </ul>	<ul style="list-style-type: none"> <li>· Regressie test is geslaagd en/of bevindingen geaccepteerd</li> </ul>

## 4.5 Test tools

Dit hoofdstuk gaat over de tools die gebruikt worden met betrekking tot testen in het 112 NL project.

**JIRA** zal gebruikt worden voor het beheren van requirements en bevindingen. JIRA is de enige bron van 'waarheid'. Als requirements (non-functionel and functional) niet in JIRA zijn omschreven, dan kan het ook niet getest worden.

- Bij elke stories zijn test cases toegevoegd in de taak Testvoorbereiding. Alle testresultaten zijn te vinden in de Task voor oplevering en eventueel Technical tasks die zijn aangemaakt nav het testen. Overzicht van de testresultaten wordt bijgehouden in Excel.
- Alle bevindingen worden gemanaged in JIRA.
- Alle communicatie over requirements en bevindingen gebeurt via JIRA.
- Alleen QA kan een bevinding sluiten. De gesloten bevinding of story wordt daarna geassigned aan de BA.

**Confluence** bevat alle documentatie rondom test, user stories, deployment, go live documenten etc. Voor testen betekent dit:

- Beheren van manuele testcases en automatische testcases
- Uitvoeren van de testcases en het volgen van uitvoeringsproces
- Opslaan van test resultaten
- Tracken van testcases en requirements, zodat progressie en coverage bepaald kan worden
- Test rapporten
- Run automated tests

**Postman** is een Chrome extentie om een HTTP request te maken. Dit zal gebruikt worden door developers en test zodat zij kunnen debuggen wat is verstuurd en ontvangen door een API.

**AML HTTPS testtool** is een testtool die emergency calls naar de AML HTTPS POST Service kan simuleren.

**Charles proxy** is een HTTP proxy. Charles Proxy kan het verkeer van Android en iOS devices naar de API afvangen, waardoor developers en testers dit verkeer kunnen debuggen. De tester kan door middel van Charles Proxy ook service errors forceren.

**Robotframework** is een automation tool die voor zowel native iOS, native Android en API test automation gebruikt kan worden. Eclipse is de tool waarin testscripts worden geschreven voor Robotframework. In dit project zullen we Robotframework gebruiken in combinatie met:

- **Appium** is een server die voor functional automated tests fungeert voor native iOS en native Android
- **Requests & jsonlibrary** zijn de libraries die gebruikt worden om automated REST en SOAP API tests te schrijven en uit te voeren
- **WebSocketClient** is een library die word gebruikt om automated API voor websockets te testen

**Gitlab** CI wordt gebruikt om de automatische testscripts te beheren en fungeert als CI/CD pipeline om automatische tests te runnen en de resultaten weg te schrijven.

## 112 NL

Test plan – Team App (DPC)

**Graylog** is de logging database waarnaar de testresultaten worden geschreven. Via Grafana (dashboard interface) zijn de testresultaten per dag/ per sprint zichtbaar en transparant voor het development team.

**Google Firebase** is de tool om de iOS en Android app te monitoren met betrekking tot performance en gebruikersaantallen.

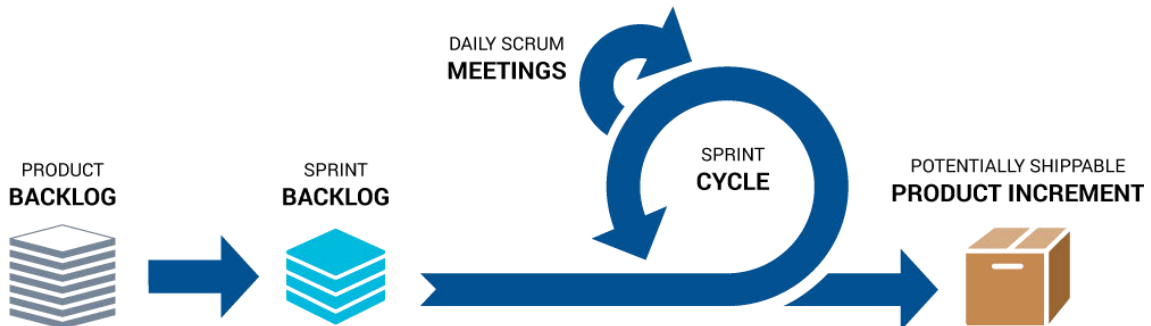
**Google Crashlytics** is de tool om de iOS en Android crashes te bekijken.

# 112 NL

Test plan – Team App (DPC)

## 5 TEST BENADERING

Voor 112 NL project wordt agile ontwikkeld en dus continue getest. De volgende afbeelding verduidt het werkritme voor het App Team binnen het 112 NL project. Het App Team werkt in cycles van tweewekelijkse sprints.



Testactiviteiten voor de start van de sprint voor QA:

- Voorbereiden van testomgevingen
- Voorbereiden van test data
- Creëren van manuele en/ of automatische testcases
  - *API, functioneel (end-to-end), regressie*

Testactiviteiten gedurende de sprints voor QA:

- Deelnemen aan sprintplanning, refinement en scoping
- Uitvoeren van manuele en/ of automatische tests
  - *API, functioneel (end-to-end)*
- Design checks indien nodig
- Testresultaten bijhouden
- Bevindingen tonen
- Hertesten van gevonden bevindingen
- Rapporteren op test progressie en bevindingen
- Deelnemen aan demo en retrospective meetings.
- Onderhouden van automatische tests

Testactiviteiten voor Go-live

- Regressietest op acceptatie omgeving (Regressie test)
  - *Regressie test*
- Regressietest op productie omgeving (Regressie test)
  - *Regressie test*
- Release management

### 5.1 Test Deliverables

Nodig voor testvoorbereiding en testuitvoering:

- Test strategie document

# 112 NL

## Test plan – Team App (DPC)

- Test Data (wanneer nodig)

Volgt na uitvoering tests:

- Sprint report
- Vrijgave advies (voor go live)
- Release checklist (voor go live)

## 5.2 Test devices

Om te testen voor 112 NL zijn er devices, in dit geval telefoons, nodig met de 112 NL app geïnstalleerd. Tests worden uitgevoerd op zowel iOS en Android smartphones. Ook developers gebruiken regelmatig fysieke devices om te developen en te debuggen. De developers maken tevens gebruik van simulators. De volgende test toestellen worden door het App Team gebruikt om zo volledig mogelijk te testen:

Android			
Device	OS	Resolutie	Size
Samsung S6 - 2	7.0	2560 x 1440	5.1"
Nexus 6p	8.1.	2560 x 1440	5.7"
Huawei P20 Pro	8.1.	2240 x 1080	6.1"
Google Pixel	10.0	1440 x 2960	6.3"
Samsung Galaxy A8	9.0	2220 x 1080	5.6"
Samsung Galaxy A5 (2016)	7.0	1080 x 1920	5.2"
Nexus 5x	6.0	1920 x 1080	5.2"
Samsung Galaxy S9	10.0	2960 x 1440	5.8"
Samsung Galaxy S8	10.0	2960x1440	5.8"
Samsung Galaxy S20	10	1440x3200	6.9"
HTC one plus	9.0	2340x1080	6.41"

iOS			
Device	OS	Resolutie	Size
iPhone 5SE	11.4	640 x 1136	4"
iPhone 7	13.3.1	1334 x 750	4.7"
iPhone 8 plus	13	1920 x 1080	5.5"
iPhone X	13	2436 x 1125	5.8"
iPhone Xs Max	13	2688 x 1242	6.5"
iPhone 11	13	2436 x 1125	5.8"

\*Let op! Deze lijst wordt voor elke go live geupdate aan de hand van nieuwe releases van OS versies en devices.

## 112 NL

Test plan – Team App (DPC)

In scope zijn telefonische devices van Android en iOS, dus geen tablets. Onderstaande cijfers gaan over mei 2020 in Nederland. De volgende OS versies van beide platformen zijn in scope voor 112NL:

Android		iOS	
OS	Gebruikers	OS	Gebruikers
10	37%	13	80%
9	31%	12	10%
8	17%	10	2%
7	5%	9	2%
6	4%		
<a href="https://gs.statcounter.com/android-version-market-share/mobile-tablet/netherlands">https://gs.statcounter.com/android-version-market-share/mobile-tablet/netherlands</a>		<a href="https://gs.statcounter.com/os-version-market-share/ios/mobile-tablet/netherlands">https://gs.statcounter.com/os-version-market-share/ios/mobile-tablet/netherlands</a>	

\*Deze cijfers gaan over zowel telefoons als tablets.

\*\* Beslissing ligt nog aan de kant van LMS, zodra een beslissing over de scope van OS versies is genomen zal er een nieuwe versie van het Testplan komen.

### 5.3 Test data

Voor de System Test en UI test is er voor de volgende onderdelen test data nodig:

- Websocket berichten
- Foto's en of video's
- Persoonlijke gegevens
- Locatie

Er wordt geen productiedata gebruikt voor testdoeleinden.



## 112 NL

Test plan – Team App (DPC)

### 6 TEST OMGEVINGEN

De volgende testomgevingen zullen bereikbaar zijn tijdens de ontwikkeling van 112 NL

APP	GMS API	Comment
DEV	TEST	Deze omgeving is voor development
QA	TEST	Deze omgeving is voor interne tests
ACCEPTATIE	ACCEPTATIE	Deze omgeving is voor de acceptatie tests en UAT
PRODUCTIE	PRODUCTIE	Deze omgeving is voor de (eventuele) pilots en de eindgebruiker na go live

## 7 TESTMANAGEMENT & RAPPORTEREN

### 7.1 Test rapport

Dit hoofdstuk beschrijft hoe en naar wie het testrapport gerapporteerd moet worden.

**Release test rapport** bestaat uit opgeleverde userstories inclusief testcase en resultaat. Eventueel is een lijst met bevindingen en bugs toegevoegd. Dit rapport bestaat uit het vrijgaveadvies en restrisico's die uit de releasechecklist naar voren zijn gekomen en in overleg met PO.

Communicatie: Confluence en e-mail

Personen: Product Owner, Business Owner, Project Manager, Scrum Master

**Sprint status rapport** bestaat uit de stories van de sprint backlog, eventuele (openstaande) bevindingen en velocity. Voor het 112NL project wordt de sprint status raport gemaakt door de business analyst

Communicaite: confluence en e-mail

Personen: Product Owner, Business Owner, Project Manager, Scrum Master

### 7.2 Bugs en bevindingen management

Tijdens het testen kunnen issues worden geïdentificeerd als:

- De resultaten van de test niet zijn zoals verwacht
- Als een test niet kan worden uitgevoerd

Het development team streeft er naar zo min mogelijk bugs mee te nemen naar volgende sprints. Het doel van het development team is om gevonden issues gedurende een sprint in dezelfde sprint op te lossen. Mochten er bevindingen zijn die niet tijdens de sprint kunnen worden opgelost, dan zal de BA dit in overleg met PO/BO op de backlog plaatsen om direct te prioriteren. Hetzelfde geldt voor bugs gevonden in productie.

### 7.3 Defect classificatie

Defect classificatie betreft bevindingen op productie en gebeurt in overleg tussen QA en BA, op advies van het development team. Mocht het defect een hoog risico hebben, dan zullen de BO/PO geïnformeerd worden.

Prioriteit	Beschrijving
Blocker	Het systeem is onbereikbaar of een kritiek process of functie is kapot.

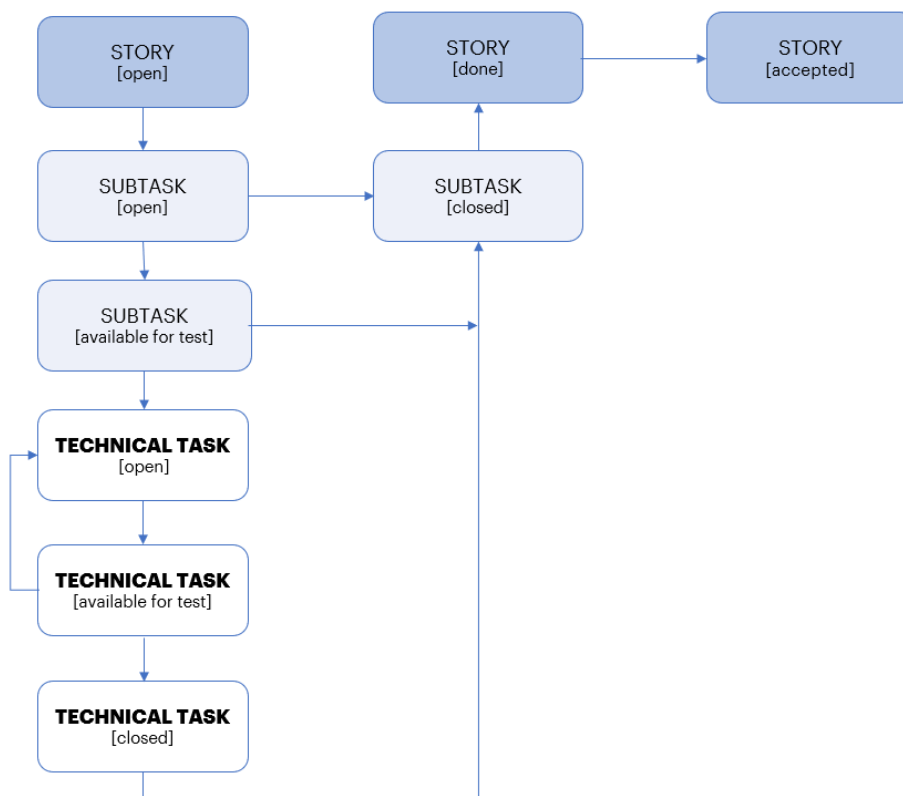
## 112 NL

Test plan – Team App (DPC)

Critical	Een kritieke functie is kapot maar er is een workaround. Het probleem moet in de eerstvolgende build of hotfix worden opgelost.
Major	Een niet kritieke functie is kapot maar er is een workaround. Het probleem moet worden opgelost voor de eerstkomende release.
Minor	Een kleine functie werkt niet meer en een oplossing is makkelijk door te voeren. De oplossing wordt besproken en ingepland.
Trivial	Een esthetische bevinding die de gebruiker niet verhindert in het gebruik van functionaliteiten. De oplossing wordt besproken en ingepland.

### 7.4 Story bug flow

Onderstaand schema laat zien hoe de flow binnen het team gaat met betrekking tot een bevinding uit een opgeleverde story.

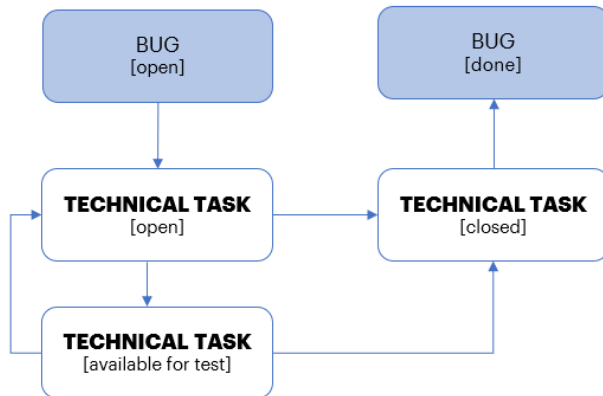


### 7.5 Bug flow

Onderstaand schema laat zien hoe de flow binnen het team gaat met betrekking tot een bug uit productie, of een bug gevonden door iemand anders uit het team of daarbuiten.

## 112 NL

Test plan – Team App (DPC)

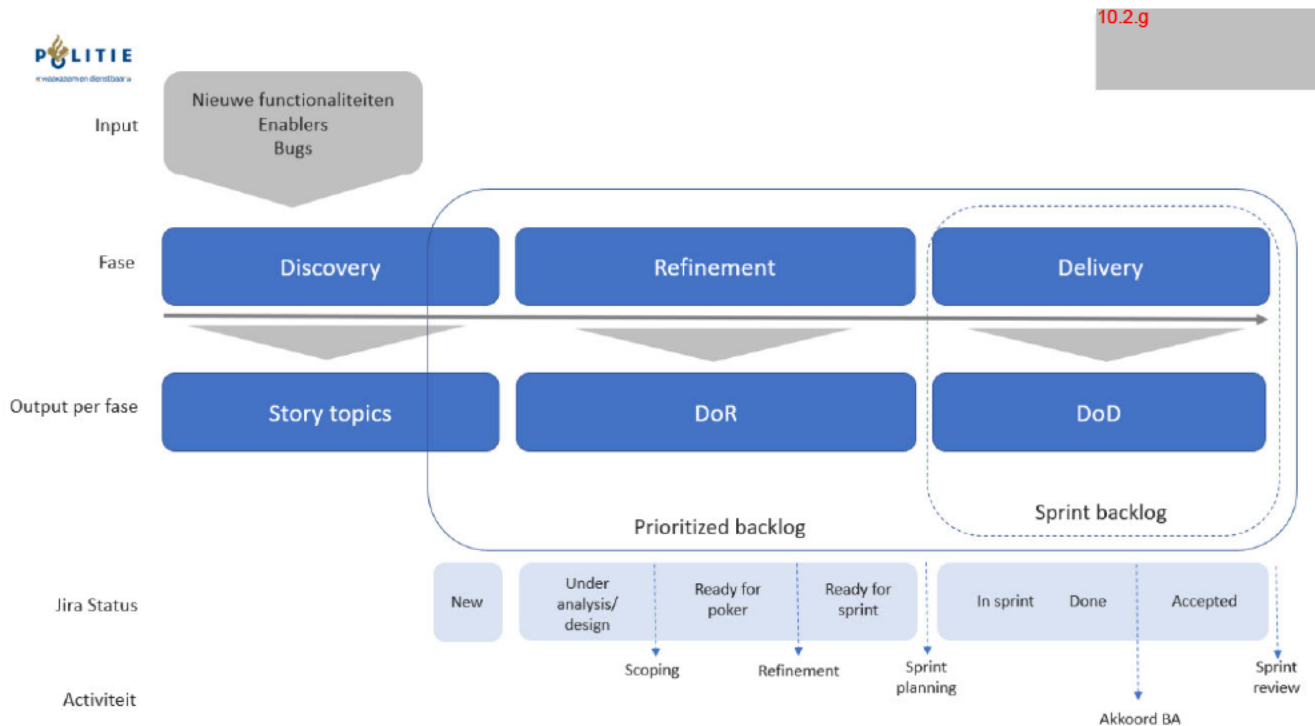


### 7.6 Risico's

'No-risk, no test' is het uitgangspunt, er wordt aldus risico gebaseerd getest. Zie GAQA, paragraaf 2.1, voor een beschrijving. Om inzicht in de risico's te krijgen, wordt er elke sprint refinements gehouden waar user stories worden besproken en eventuele risico's (h)erkend.

## 8 TEST ACTIVITEITEN EN SCHEMA

Dit hoofdstuk omschrijft de algemene werkschema van het scrumteam. Onderstaand schema is de Way of Working van het development team.



- Testcases worden geschreven voordat de refinement plaatsvindt
- Aan het begin van de sprint worden de testcases van de sprintbacklog gereviewd door BA, SA en een backend developer.
- Testresultaten worden gedeeld via het Grafana dashboard, Jira en Confluence
- Nadat development taken heeft opgeleverd worden deze geassignd aan test. Na test uitvoer wordt de taak gesloten, of blijft deze open en een technical task aangemaakt. Zie story bug flow voor een schematische weergave.
- Als alle taken in een story op Done staan. Dan wordt de gehele story op Done gezet en geaccepteerd.

## 9 ROLLEN EN VERANTWOORDELIJKHEDEN

De volgende tabel toont welke personen onderdeel zijn van het team voor de delivery voor 112 NL app en backend services:

Naam	Rol/verantwoordelijkheid	Taak
10.2.e	10.2.e	<ul style="list-style-type: none"> <li>• Reviewed en accepteer de Test Strategie</li> <li>• Zorgt voor support vanuit de business met betrekking tot requirements en sign-off activiteiten, prioritering van taken en bevindingen.</li> </ul>
10.2.e	Business Analyst	<ul style="list-style-type: none"> <li>• Maken van user stories</li> <li>• Review Test Strategie en test cases</li> <li>• Support voor requirements vanuit de business.</li> <li>• Prioritesing van de backlog in overleg met BO</li> </ul>
10.2.e	Tester	<ul style="list-style-type: none"> <li>• Test strategie</li> <li>• Test management</li> <li>• Testvoorbereiding</li> <li>• Test uitvoering</li> <li>• Release management development team</li> <li>• Design check</li> </ul>
10.2.e	iOS developer	<ul style="list-style-type: none"> <li>• Development iOS application</li> </ul>
10.2.e	Android developer & backend developer	<ul style="list-style-type: none"> <li>• Development Android application</li> <li>• Development backend servers</li> </ul>
10.2.e	Android developer	<ul style="list-style-type: none"> <li>• Development Android application</li> </ul>
10.2.e	Backend developer	<ul style="list-style-type: none"> <li>• Development backend servers</li> <li>• CI/CD opzetten en beheren</li> </ul>

## 10 APPENDIX

Tijdens de oplevering van app en backend services voor **112 NL** project zal de focus liggen op de volgende testtypes om de kwaliteit te kunnen waarborgen:

**Regressie Test** - het doel is om te testen of later gereleasde code impact heeft gehad code die eerder is geschreven en getest.

**Smoke Test** – een selectie van tests zullen uitgevoerd worden om te verzekeren dat het basis systeem voldoet aan de requirements en dat er geen onverwachtse bugs

**Performance Test** - Testing performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage.

**Non-functional Test** - Non-functional testen zijn testen die niet te maken hebben met functionaliteiten in een app of service, maar wel nodig zijn voor optimale werking van de functionaliteiten voor iedereen.

**Accessibility Test** – is een test die valideert of de UI voldoet aan de toegankelijkheid voor mensen met een verminderend zicht, of andere issues waardoor het gebruik van een app niet vanzelfsprekend is.

**Security Test** – heeft als doel om gebreken in een systeem aan het licht te brengen waardoor gegevens niet beschermd kunnen worden.

**UI Test** - UI Testen zijn testen die gaan over User Interface functionaliteiten: onderdelen van het systeem die zichtbaar zijn voor een gebruiker.

**Functional Test** – test of een onderdeel van het systeem voldoet aan de functionele requirements.

**System Test** - zijn testen die het complete geïntegreerde systeem testen of zij voldoen aan de requirements. Met name connectiviteit met API's en User Interface zijn hierin van belang.

**Integration Test** - Integration test is het testen van één groep van componenten.

**Unit Test** - Unit testen zijn testen die componenten in de code testen.

**User Acceptance Testing (UAT)** – The aim is to demonstrate that the integrated system conforms to requirements and design. The aim being to ensure business acceptance that the new system with the enhancements produces results as expected.