

# Testplan

Databasefactory 5



15-06-2017  
Arnhem

# Informatieblad

ISE-Project Omgevingswet

Namens de Hogeschool van Arnhem en Nijmegen uitgevoerd voor CGI Nederland.

Het project wordt uitgevoerd door Database Factory 5.

Projectleden:

- Michiel Bos 561485
- Ricardo van Burik 569129
- Johan Heij 584299
- Duncan Luiten 553272
- Tristan de Roo 563431

Begeleiders:

- Tim de Goede
- Pim Haenen

Datum: 15 juni 2017

Locatie: Ruitenberglaan 31, 6826 CC Arnhem

Versie: 2.0

## Versiebeheer

Versie	Datum	Aanpassingen
1.0	19 mei 2017	
2.0	15 juni 2017	<ul style="list-style-type: none"><li>• Paar kleine dingen aangepast, naar aanleiding van eerder gehouden review.</li></ul>

## Inhoudsopgave

<b>1. Inleiding</b>	4
<b>2. Opdrachtformulering</b>	5
2.1. Binnen scope	5
2.2. Buiten scope	5
2.3. Randvoorwaarden	5
<b>3. Testbasis</b>	5
<b>4. Teststrategie</b>	6
4.1. Productrisico-analyse	6
4.2. Testsoorten	7
<b>5. Testaanpak</b>	8
5.1. Testproducten	8
5.2. Reviews	8
5.3. Testuitvoering	9
<b>6. Tooling en beheer</b>	9

# 1. Inleiding

Het testplan beschrijft de wijze waarop Database factory 5 omgaat met het testen van producten. Dit testplan heeft betrekking op de database, de beheerapplicatie, en de webapplicatie.

Er wordt aandacht besteed aan de testscope en de randvoorwaarden. De testbasis beschrijft op basis waarvan de testcases worden afgeleid. Daarna komt de teststrategie aan de orde, waarin de productrisico-analyse en de testsoorten zijn beschreven. Vervolgens wordt er een overzicht gegeven van de testaanpak. Tot slot wordt er nog een en ander uitgelegd met betrekking tot tooling en beheer.

## 2. Opdrachtformulering

In dit hoofdstuk wordt aandacht besteed aan welke testactiviteiten binnen de scope vallen en welke buiten de scope vallen. Daarnaast worden de randvoorwaarden genoemd die nodig zijn om ervoor te zorgen dat alle tests met succes afgerond kunnen worden.

### 2.1. Binnen scope

Binnen de scope van de testopdracht valt het testen van sommige database gerelateerde constraints. Denk hierbij aan het testen van check constraints, triggers, functions en stored procedures.

Daarnaast vallen de volgende zaken ook binnen de scope:

- Testen van de hosting compatibiliteit van de database met Windows 7.
- Testen van de database constraints in SQL Server 2012 en 2014.
- Testen van de front-end van de webapplicatie in Chrome, Firefox en Edge op Windows 7, 8.1, 10, Linux (in ieder geval Ubuntu 16.04), en eventueel Android (2.6 en hoger) en iOS (10.0 en hoger).
- Testen van de front-end van de beheerapplicatie in Windows 7, 8.1 en 10.

### 2.2. Buiten scope

Buiten de scope valt onder andere het testen van primary keys, alternative keys, en foreign keys. Dit zijn relatief eenvoudige constraints, waarvan wij het niet nodig achten om deze apart te testen.

Daarnaast vallen de volgende zaken ook buiten de scope:

- Testen van hosting compatibiliteit van de database met systemen anders dan genoemd.
- Testen van database constraints in systemen anders dan genoemd.
- Testen van de front-end in systemen of software anders dan genoemd.

### 2.3. Randvoorwaarden

De voorwaarden die nodig zijn om goed te kunnen testen, zijn als volgt:

- Werkende broncode om alle gemaakte producten na te kunnen bouwen.
- Besturingssystemen die voldoen aan de eerder genoemde scope.
- Internetbrowsers die voldoen aan de eerder genoemde scope.
- Databasesoftware die voldoet aan de eerder genoemde scope.
- Een werkende internetverbinding.

## 3. Testbasis

De testcases worden afgeleid uit een aantal eerder opgeleverde producten. Denk hierbij aan het acceptatieplan, de use cases en de navigation map.

## 4. Teststrategie

Bij het opstellen van testcases moet rekening gehouden worden met een aantal aspecten. Zo kan voor een testcase bijvoorbeeld een bepaalde snelheid vereist zijn, of moet de gebruiksvriendelijkheid of veiligheid getest worden. Deze zaken worden op een rij gezet in hoofdstuk 4.1. waarin een analyse wordt gegeven van de risico's per testproduct. Verder komen in hoofdstuk 4.2 de verschillende testsoorten aan bod.

### 4.1. Productrisico-analyse

Kenmerk/onderdeel	Risicoklasse	Argumentatie
Gebruiksvriendelijkheid beheerapplicatie	Laag	Er is geen eis gesteld aan gebruiksvriendelijkheid, omdat de producten na oplevering nog verder zullen worden ontwikkeld.
Gebruiksvriendelijkheid webapplicatie	Laag	Er is geen eis gesteld aan gebruiksvriendelijkheid, omdat de producten na oplevering nog verder zullen worden ontwikkeld.
Beveiliging beheerapplicatie	Middel	De beheerapplicatie wordt na oplevering nog verder ontwikkeld. Ondanks dit feit is de beveiliging toch van enig belang, omdat vanuit de beheerapplicatie per ongeluk of met opzet tot op zekere hoogte geknoeid kan worden met de database.
Beveiliging webapplicatie	Middel	De webapplicatie wordt na oplevering nog verder ontwikkeld. Ondanks dit feit is de beveiliging toch van enig belang, omdat vanuit de webapplicatie per ongeluk of met opzet tot op zekere hoogte geknoeid kan worden met de database.
Beveiliging/data-integriteit database	Hoog	De beveiliging van de database is van groot belang. Er worden persoonsgegevens in opgeslagen die bij wet goed beschermd <i>moeten</i> worden. Ook de data-integriteit is van belang, met name om de database toekomst-bestendig te krijgen en te houden.
Performance beheerapplicatie	Laag	Er is geen eis gesteld aan de performance van de beheerapplicatie, omdat de producten na oplevering nog verder zullen worden ontwikkeld.
Performance webapplicatie	Laag	Er is geen eis gesteld aan de performance van de webapplicatie, omdat de producten na oplevering nog verder zullen worden ontwikkeld.

Performance database	Hoog	De performance van de database is van belang. De database is het centrum, waar zowel de beheer- als de webapplicatie gebruik van maken.
Use case “starten van een project”	Hoog	Deze use case is het belangrijkste onderdeel van het systeem. Zonder het aanmaken van een project is er geen aanleiding om de overige use cases uit te voeren.
Use case “toekennen van een coördinator aan een project”	Hoog	Een project kan niet zonder coördinator gestart worden. Gaat er hier iets mis, dan kan de rest van het gehele proces niet doorlopen worden.
Use case “beheren van belanghebbenden”	Laag	Het beheren van belanghebbenden is iets dat niet van groot belang is.
Use case “uploaden van projectinformatie”	Hoog	Deze use case is een van de belangrijkste onderdelen van het systeem. Zonder goede informatievoorziening wordt het immers moeilijk om vergunningen te verlenen of om bezwaar te maken.
Use case “toekennen van een bevoegd gezaghebber”	Middel	De bevoegd gezaghebber moet normaal gesproken de vergunningscheck uitvoeren, en moet dus bekend zijn. Deze kan echter, in noodgevallen, via de database toegevoegd worden.
Use case “toevoegen van een vergunningsaanvraag”	Hoog	Het toevoegen van een vergunningsaanvraag is van belang omdat enkele processen er op voortbouwen.
Use case “behandelen van een vergunningsaanvraag”	Hoog	Een project mag pas daadwerkelijk van start gaan wanneer alle vergunningsaanvragen zijn goedgekeurd.
Use case “abonneren op een project”	Middel	Niet per sé vereist in de applicatie, omdat er een globaal overzicht getoond moet worden, maar is wel fijn voor het gebruikersgemak.
Use case “maken van bezwaar”	Hoog	Het is belangrijk dat gebruikers hun stem kunnen laten horen.
Use case “behandelen van bezwaar”	Laag	Dit is een belangrijk onderdeel, maar kan in noodgevallen via de database gedaan worden.
Use case “stopzetten van een project”	Laag	Dit is een belangrijk onderdeel, maar kan in noodgevallen via de database gedaan worden.

## 4.2. Testsoorten

Hieronder worden de verschillende testsoorten beschreven alsook welk doel deze dienen.



Testsoort	Beschrijving	Doel
Bouwtest	Testsoort die door programmeurs wordt uitgevoerd. Hierbij wordt zowel gebruik gemaakt van Unit Tests als andere testtechnieken.	Verifiëren dat de code testgereed is en dat aangebrachte wijzigingen geen ongewenste neveneffecten hebben gehad.
Systeemtest	Testsoort die door testers wordt uitgevoerd. Aan de hand van Use Cases en acceptatiecriteria uit het Acceptatie Plan worden testgevallen uitgewerkt en uitgevoerd.	Metten en daarmee helpen verbeteren van de kwaliteit van code en andere werkproducten, verifiëren dat de gerealiseerde software voldoet aan de overeengekomen specificaties en beoordelen van de oplevergereedheid van nieuwe en eerder gerealiseerde functionaliteit.
GAT ondersteuning	Testsoort die door gebruikers en andere domeindeskundigen wordt uitgevoerd. Het testteam ondersteunt deze testinspanningen.	Beoordelen of de gerealiseerde functionaliteit voldoet aan de bedoeling die de domeindeskundige voor ogen stond bij het formuleren van de specificaties.

## 5. Testaanpak

In dit hoofdstuk wordt beschreven hoe de testaanpak eruit ziet. De testproducten worden genoemd en toegelicht. Reviews die hierop van toepassing zijn worden uitgelegd. Tot slot wordt beschreven op welke manier de tests precies uitgevoerd worden.

### 5.1. Testproducten

Werkproduct	Toelichting
Testscript	Het testscript wordt uitgevoerd ten behoeve van het geautomatiseerd testen. Dit komt binnen alle iteraties van de constructie fase en de transitie fase aan bod.
Testrapport	In het testrapport worden de verschillende testcases behandeld, en wat de resultaten van deze testen zijn. Dit komt vlak voor de eindoplevering aan de orde.

### 5.2. Reviews

Naast het uitvoeren van de tests zelf, moet ook de testcode gewaarborgd worden. Immers, als de tests slagen terwijl de testcode zelf van onvoldoende kwaliteit is, kan er net zo goed

niet getest worden. Bij het opleveren van een nieuw stuk testcode moet die eerst gereviewd worden.

### 5.3. Testuitvoering

Wanneer een nieuwe constraint af is, moet deze eerst getest worden voordat die in de primaire database komt te staan. Deze tests moeten vooraf uitgevoerd worden in een testdatabase, maar dit kan ook op een lokale database gedaan worden.

## 6. Tooling en beheer

De testtools die gebruikt worden zijn sql testscripts die min of meer ge-automatiseerd getest kunnen worden. De beheerapplicatie wordt automatisch getest door middel van unit testen. De performance wordt getest door enkele resource-heavy queries uit te voeren, terwijl de stresstest wordt uitgevoerd door middel van script-kiddie software. De database wordt automatisch gevuld door middel van de software "RedGate SQLDataGenerator". De resultaten van de uitgevoerde tests worden behandeld in een testrapport.