



IPBEV – Testplan

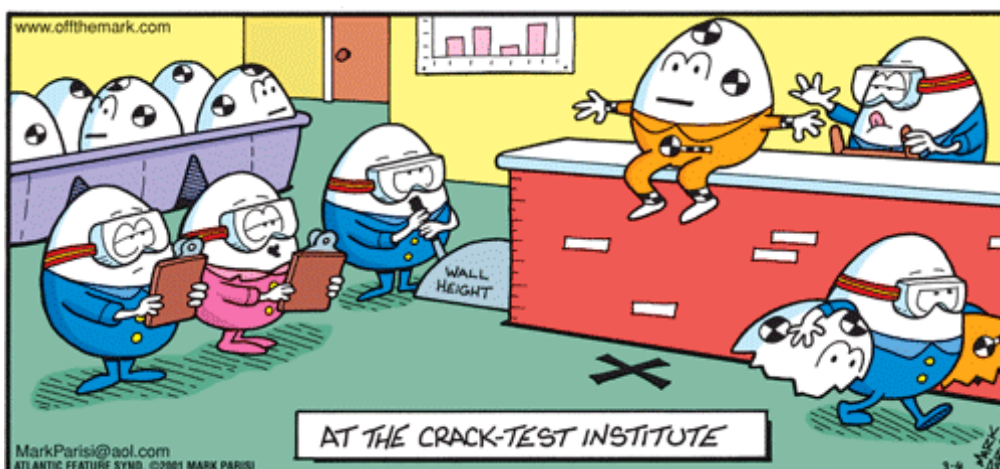
Hogeschool Leiden - Informatica

Voor Stichting Wireless Leiden waren wij als student ingezet om een interactieve nodemap te maken om de verschillende nodes van Wireless Leiden te kunnen tonen aan gebruikers en beheerders.

Pieter Naber
17-4-2010

Inhoudsopgave

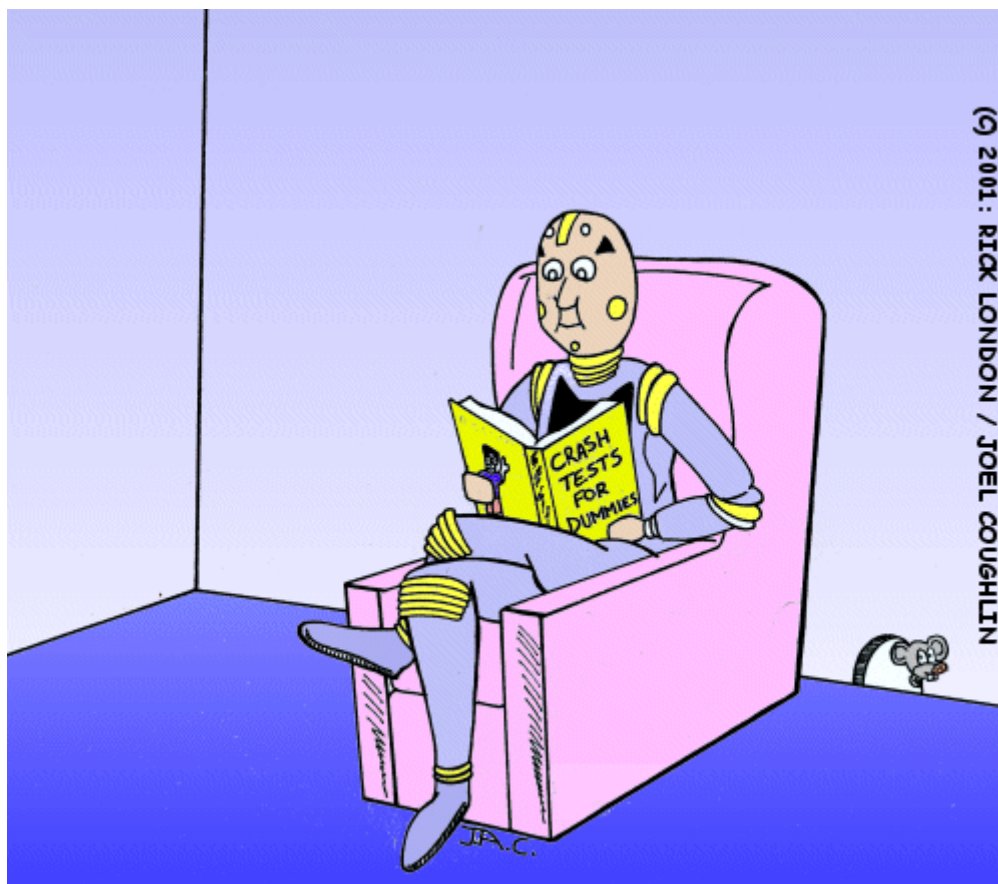
1	Inleiding	3
2	Test Risico Analyse (TRA)	4
2.1	Input parser	4
2.2	KML parser.....	5
2.3	Relatief belang.....	6
2.4	Het resultaat.....	6
3	Logische Testontwerp (LT)	7
3.1	Input parser – Lees locatie data – Lees “name”	7
3.2	Input parser – Lees locatie data – Lees “location”	7
3.3	Input parser – Lees locatie data – Lees “latitude”	7
3.4	Input parser – Lees locatie data – Lees “longitude”	7
3.5	Input parser – Lees status data – Lees “host_name”	7
3.6	Input parser – Lees status data – Lees “current_state”	8
3.7	Input parser – Lees interlinks – Lees “ip”	8
3.8	Input parser – Lees interlinks – Lees “netmask”	8
3.9	KML parser – Maak “Points” – Parse lokale variabelen	8
	KML parser – Maak “Points” – Parse globale variabelen	9
3.10	KML parser – Maak “LineString” – Parse lokale variabelen	9
3.11	KML parser – Maak “LineString” – Parse globale variabelen	9
3.12	KML parser – Link naar bestand – Cache bestanden.....	9
3.13	KML parser – Link naar bestand – Link naar laatste KML bestand.....	10
4	Tot slot	11



1 Inleiding

Als je voor een project van een redelijke omvang een logisch testplan gaat schrijven begrijp je ineens hoeveel testen je kan bedenken voordat je naar je eigen idee klaar bent. En zelfs dan... Ben je niets vergeten? Een dilemma... Wanneer ga je door... Wanneer ben je klaar? Door de methode van TestGoal te gebruiken heb je enigszins een leidraad, maar het dilemma blijft staan.

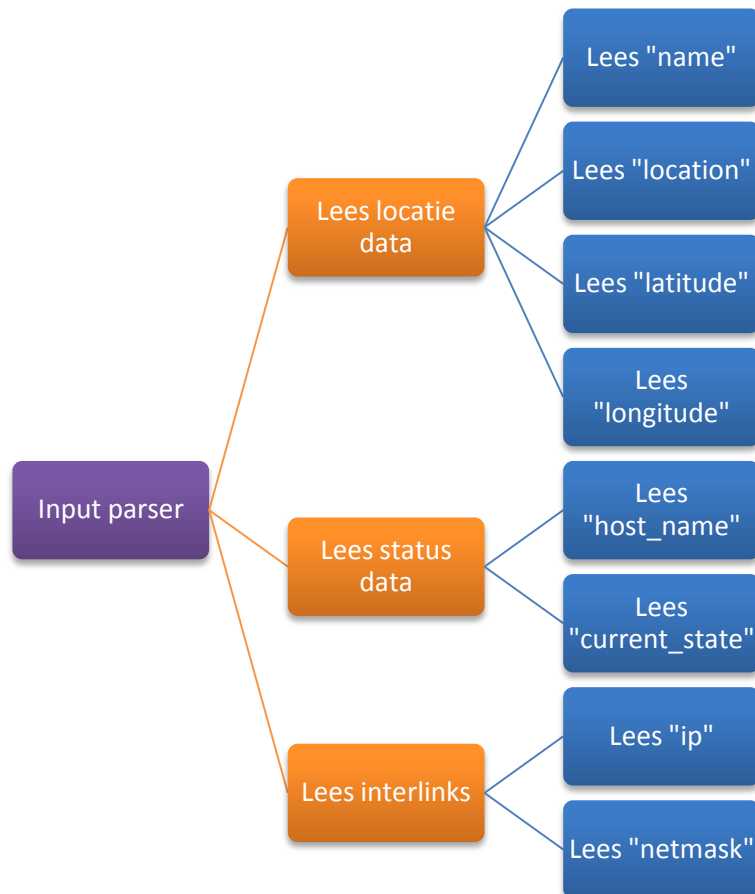
In dit testplan beschrijven we het logische testplan van de nodemap applicatie die ik heb gemaakt voor het project IPBEV: Informatica Project Beveiliging. Tijdens het programmeren ben ik alleen bezig geweest met de back-end (het maken van een KML bestand vanuit input bestanden) en voor dat deel van de applicatie zal ik de tests beschrijven. De front-end is onze interpretatie van het KML bestand en is makkelijk aan te passen door gebruikers die iets anders willen. Deze zal daarom niet worden meegenomen in deze tests.



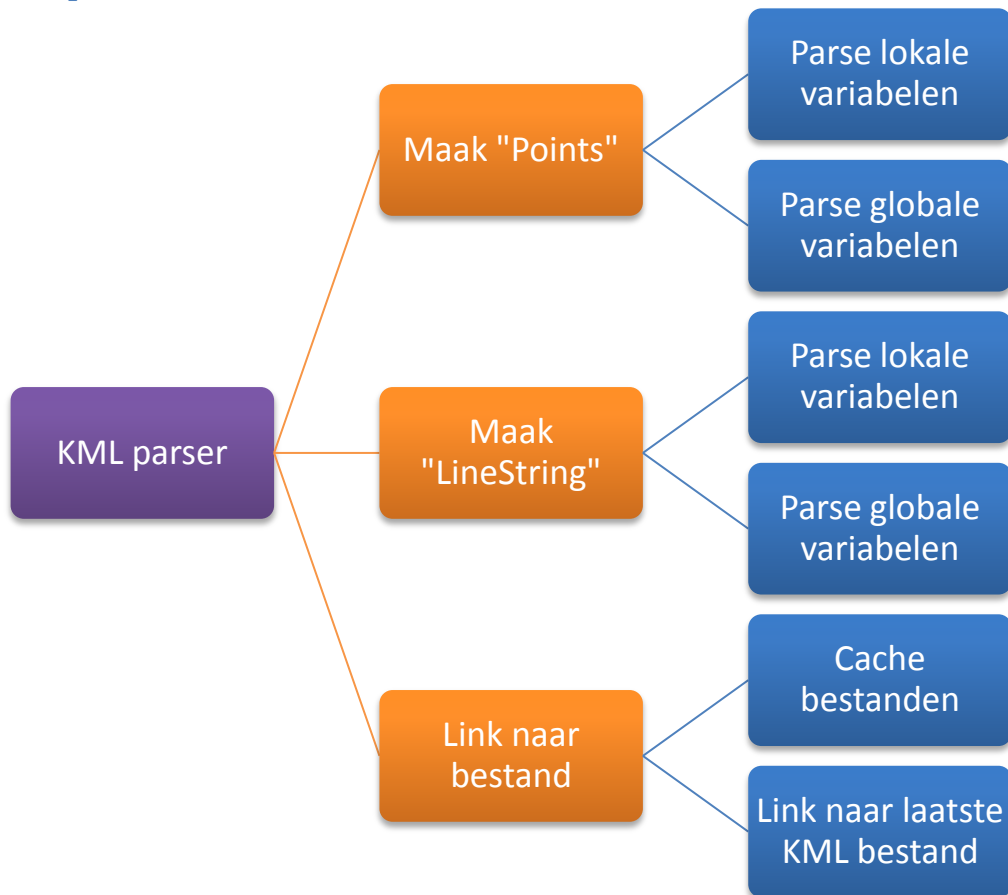
2 Test Risico Analyse (TRA)

Om te komen tot het logische testplan, heb ik er voor gekozen om te beginnen met een Test Risico Analyse om voor mezelf een beeld te krijgen van de te testen functies.

2.1 Input parser



2.2 KML parser



2.3 Relatief belang

Nadat de clusters zijn bepaald, ga ik kijken naar het relatief belang. Bij "Verdeling 1" maak ik een schatting wat de opdrachtgever zal vinden, gebruik makend van de gesprekken met de opdrachtgeverde informatie in het bijgeleverde document. Bij "Verdeling 2" geef ik een verdeling uitgaande van mijn eigen ervaring.

Risicogebied:	Verd. 1:	Verd. 2:	Totaal:
Input parser – Lees locatie data – Lees "name"	9	9	18
Input parser – Lees locatie data – Lees "location"	1	1	2
Input parser – Lees locatie data – Lees "latitude"	9	9	18
Input parser – Lees locatie data – Lees "longitude"	9	9	18
Input parser – Lees status data – Lees "host_name"	1	5	6
Input parser – Lees status data – Lees "current_state"	5	9	14
Input parser – Lees interlinks – Lees "ip"	5	5	10
Input parser – Lees interlinks – Lees "netmask"	3	5	8
KML parser – Maak "Points" – Parse lokale variabelen	9	9	18
KML parser – Maak "Points" – Parse globale variabelen	9	9	18
KML parser – Maak "LineString" – Parse lokale variabelen	5	3	8
KML parser – Maak "LineString" – Parse globale variabelen	5	3	8
KML parser – Link naar bestand – Cache bestanden	3	1	4
KML parser – Link naar bestand – Link naar laatste KML bestand	9	9	18

2.4 Het resultaat

Door het beperkt aantal relatieve belangen die ik heb beschreven, zal ik niet precies de verdeling van de risicocategorieën kunnen aanhouden. Als het relatief belang even hoog is, zullen die risicogebieden in dezelfde categorie moeten. Mijn risicoanalyse komt er dan als volgt uit te zien:

Categorie:	Risicogebied:	Relatief belang:
Kritisch	Input parser – Lees locatie data – Lees "name"	18
	Input parser – Lees locatie data – Lees "latitude"	18
	Input parser – Lees locatie data – Lees "longitude"	18
	KML parser – Maak "Points" – Parse lokale variabelen	18
	KML parser – Maak "Points" – Parse globale variabelen	18
	KML parser – Link naar bestand – Link naar laatste KML bestand	18
Hoog	Input parser – Lees status data – Lees "current_state"	14
	Input parser – Lees interlinks – Lees "ip"	10
Midden	Input parser – Lees interlinks – Lees "netmask"	8
	KML parser – Maak "LineString" – Parse lokale variabelen	8
	KML parser – Maak "LineString" – Parse globale variabelen	8
Laag	Input parser – Lees status data – Lees "host_name"	6
	KML parser – Link naar bestand – Cache bestanden	4
	Input parser – Lees locatie data – Lees "location"	2

3 Logische Testontwerp (LT)

In het LT neem ik de clusters uit onze testboom en werken deze uit tot logische testgevallen. Hierbij hou je rekening met de belangrijkheid van de functies, zoals die uit TRA komen.

3.1 Input parser – Lees locatie data – Lees “name”

De volgende situaties zullen getest moeten worden:

1. De waarde “name” bestaat niet in het bestand
2. De waarde “name” is leeg
3. De waarde “name” is gevuld met enkele quotes (') en dubbele quotes (")
4. De waarde “name” is gevuld met HTML codes
5. De waarde “name” is gevuld met KML codes
6. De waarde “name” is correct gevuld

3.2 Input parser – Lees locatie data – Lees “location”

De volgende situaties zullen getest moeten worden:

1. De waarde “location” bestaat niet in het bestand
2. De waarde “location” is leeg
3. De waarde “location” is gevuld met enkele quotes (') en dubbele quotes (")
4. De waarde “location” is gevuld met HTML codes
5. De waarde “location” is gevuld met KML codes
6. De waarde “location” is correct gevuld

3.3 Input parser – Lees locatie data – Lees “latitude”

De volgende situaties zullen getest moeten worden:

1. De waarde “latitude” bestaat niet in het bestand
2. De waarde “latitude” is leeg
3. De waarde “latitude” is gevuld met enkele quotes (') en dubbele quotes (")
4. De waarde “latitude” is gevuld met HTML codes
5. De waarde “latitude” is gevuld met KML codes
6. De waarde “latitude” is correct gevuld

3.4 Input parser – Lees locatie data – Lees “longitude”

De volgende situaties zullen getest moeten worden:

1. De waarde “longitude” bestaat niet in het bestand
2. De waarde “longitude” is leeg
3. De waarde “longitude” is gevuld met enkele quotes (') en dubbele quotes (")
4. De waarde “longitude” is gevuld met HTML codes
5. De waarde “longitude” is gevuld met KML codes
6. De waarde “longitude” is correct gevuld

3.5 Input parser – Lees status data – Lees “host_name”

De volgende situaties zullen getest moeten worden:

1. De waarde "host_name" bestaat niet in het bestand
2. De waarde "host_name" is leeg
3. De waarde "host_name" is gevuld met enkele quotes (') en dubbele quotes (")
4. De waarde "host_name" is gevuld met HTML codes
5. De waarde "host_name" is gevuld met KML codes
6. De waarde "host_name" is correct gevuld

3.6 Input parser – Lees status data – Lees "current_state"

De volgende situaties zullen getest moeten worden:

1. De waarde "current_state" bestaat niet in het bestand
2. De waarde "current_state" is leeg
3. De waarde "current_state" is gevuld met enkele quotes (') en dubbele quotes (")
4. De waarde "current_state" is gevuld met HTML codes
5. De waarde "current_state" is gevuld met KML codes
6. De waarde "current_state" is correct gevuld

3.7 Input parser – Lees interlinks – Lees "ip"

De volgende situaties zullen getest moeten worden:

1. De waarde "ip" bestaat niet in het bestand
2. De waarde "ip" is leeg
3. De waarde "ip" is gevuld met enkele quotes (') en dubbele quotes (")
4. De waarde "ip" is gevuld met HTML codes
5. De waarde "ip" is gevuld met KML codes
6. De waarde "ip" is correct gevuld

3.8 Input parser – Lees interlinks – Lees "netmask"

De volgende situaties zullen getest moeten worden:

1. De waarde "netmask" bestaat niet in het bestand
2. De waarde "netmask" is leeg
3. De waarde "netmask" is gevuld met enkele quotes (') en dubbele quotes (")
4. De waarde "netmask" is gevuld met HTML codes
5. De waarde "netmask" is gevuld met KML codes
6. De waarde "netmask" is correct gevuld

3.9 KML parser – Maak "Points" – Parse lokale variabelen

De volgende situaties zullen getest moeten worden voor elke variabele apart:

1. De lokale variabele bestaat niet
2. De lokale variabele is leeg
3. De lokale variabele is gevuld met enkele quotes (') en dubbele quotes (")
4. De lokale variabele is gevuld met HTML codes
5. De lokale variabele is gevuld met KML codes
6. De lokale variabele is correct gevuld

KML parser – Maak “Points” – Parse globale variabelen

De volgende situatie zullen getest moeten worden voor de globale variabelen als geheel:

1. Het globale configuratie bestand bestaat niet
2. Het globale configuratie bestand is leeg
3. De globale variabele is correct gevuld

De volgende situaties zullen getest moeten worden voor elke variabele apart:

1. De globale variabele bestaat niet
2. De globale variabele is leeg
3. De globale variabele is gevuld met enkele quotes (') en dubbele quotes (")
4. De globale variabele is gevuld met HTML codes
5. De globale variabele is gevuld met KML codes
6. De globale variabele is correct gevuld

3.10 KML parser – Maak “LineString” – Parse lokale variabelen

De volgende situaties zullen getest moeten worden voor elke variabele apart:

1. De lokale variabele bestaat niet
2. De lokale variabele is leeg
3. De lokale variabele is gevuld met enkele quotes (') en dubbele quotes (")
4. De lokale variabele is gevuld met HTML codes
5. De lokale variabele is gevuld met KML codes
6. De lokale variabele is correct gevuld

3.11 KML parser – Maak “LineString” – Parse globale variabelen

De volgende situatie zullen getest moeten worden voor de globale variabelen als geheel:

1. Het globale configuratie bestand bestaat niet
2. Het globale configuratie bestand is leeg
3. De globale variabele is correct gevuld

De volgende situaties zullen getest moeten worden voor elke variabele apart:

1. De globale variabele bestaat niet
2. De globale variabele is leeg
3. De globale variabele is gevuld met enkele quotes (') en dubbele quotes (")
4. De globale variabele is gevuld met HTML codes
5. De globale variabele is gevuld met KML codes
6. De globale variabele is correct gevuld

3.12 KML parser – Link naar bestand – Cache bestanden

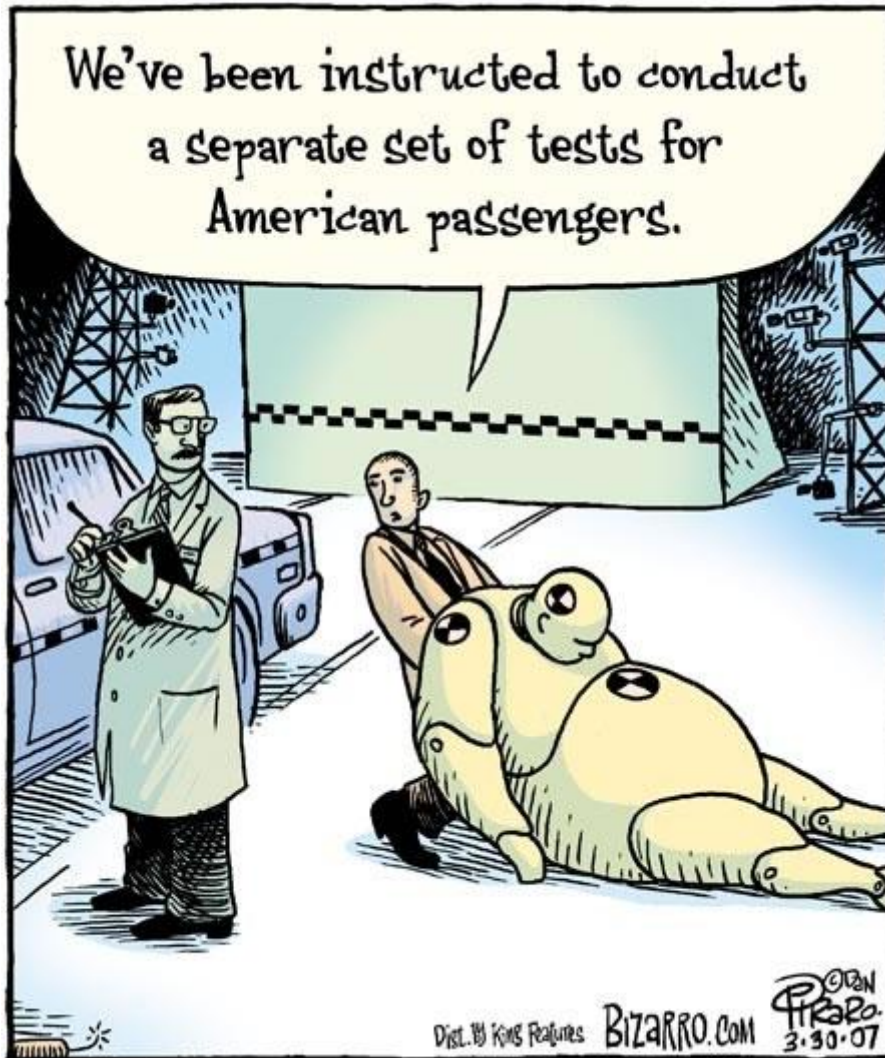
De volgende situaties zullen getest moeten worden:

1. De cache folder is niet beschrijfbaar
2. Het bestandsnaam van het cache bestand is geen correcte bestandsnaam
3. De cache folder is beschrijfbaar en het bestandsnaam is correct

3.13 KML parser – [Link naar bestand](#) – [Link naar laatste KML bestand](#)

De volgende situaties zullen getest moeten worden:

1. Er zijn nog geen KML bestanden aangemaakt
2. Er zijn meerdere KML bestanden aangemaakt



4 Tot slot

Door tijdgebrek is het helaas niet gelukt om alle tests uit te voeren in een fysiek testplan. Hopelijk kan een volgende groep wel gebruik maken van dit plan als basis voor hun tests.

