INTRAFFIC
mastering mobility

1,3 miljoen regels mission critical code omzetten naar C++, hoe test je dat?

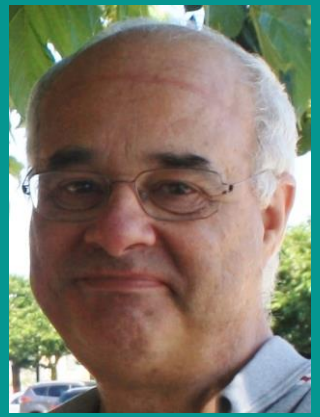
Najaarsevenement 2016

Jaap Kuilman

11 oktober 2016

Introductie

Jaap Kuilman



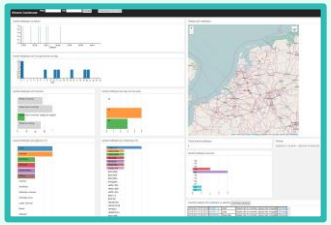
- Testconsultant bij InTraffic
- Ervaring in het OV-Domein
- Verantwoordelijk voor testen in dit project bij InTraffic
- Begeleiden testers bij het uitvoeren van het testproces

Wat doet InTraffic

Spoor

Weg

Inframonitoring



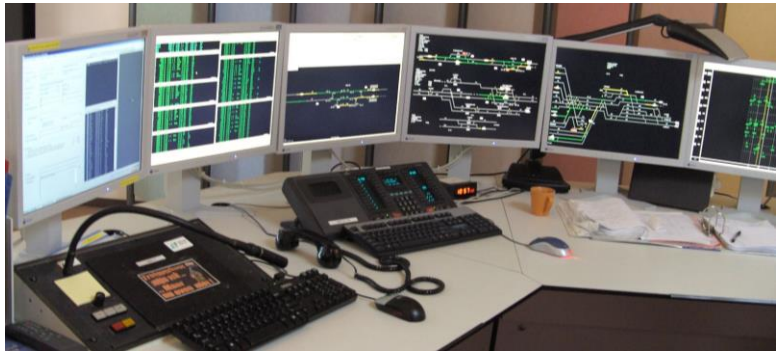
Reisinformatie

INTRAFFIC

Inhoud presentatie

- Het systeem Procesleiding Rijwegen (PRL)
- Conversie PRL van PASCAL naar C++
- Ontwikkeling van een convertertool
- Testen of de geconverteerde code voldoet
- Conclusie

Het systeem Procesleiding Rijwegen

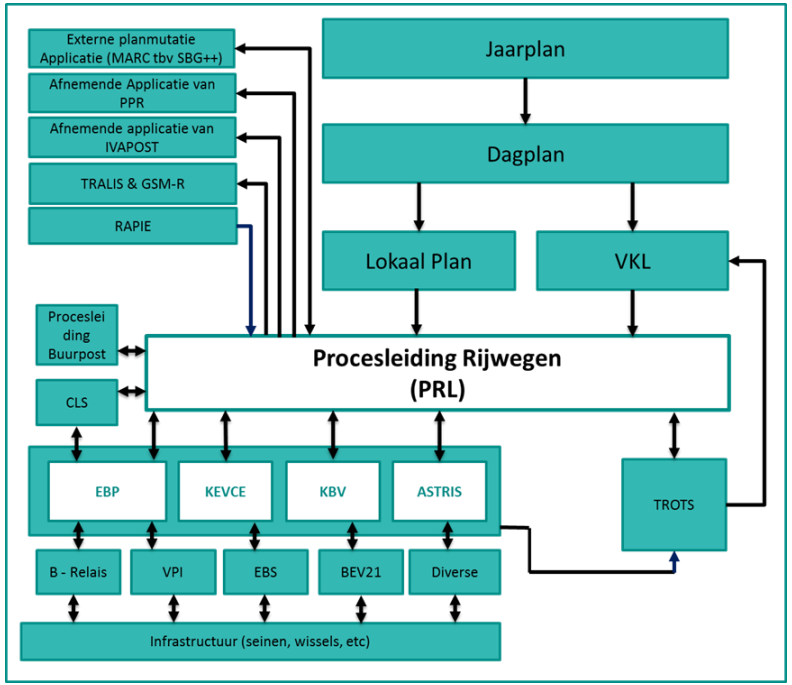


Het spoorwegnet in Nederland is druk en complex.

Procesleiding Rijwegen (PRL) stelt de rijwegen van de treinen automatisch in.

- De treindienstleiders kunnen met PRL ook handmatig seinen en wissels bedienen en handmatig rijwegen instellen voor de treinen.
- PRL helpt bij het signaleren van storingen.
- PRL draait op de 13 verkeersleidingsposten.
- PRL is een bedrijfskritisch systeem en dat vereist 24/7 beschikbaarheid.

Het systeem Procesleiding Rijwegen

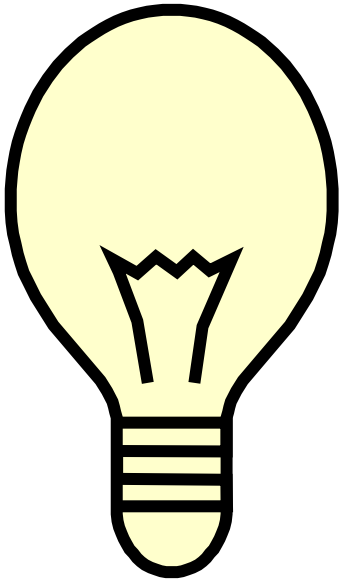


Kenmerken

- PRL bestaat uit een 50 taken (executables) die met elkaar communiceren.
- Bevat 1,3 miljoen regels code.
- Groot deel in PASCAL en een klein deel in C en C++. Het draait op VMS.
- PRL is al operationeel sinds 1994. Sindsdien wordt er continue aan doorontwikkeld.

INTRAFFIC

Conversie PRL, van PASCAL naar C++



Conversie PRL is onderdeel van het ProRail programma “Slimme Renovatie Procesleiding”

- Per 2020 hardware platform OpenVMS verlaten
  - kritische massa ontbreekt
  - kennis en kunde verdwijnt
- Conversie van Pascal naar C++.
  Een noodzakelijke stap in portering naar Linux, omdat Linux geen Pascal ondersteunt.
- Automatisering van conversie van 1,3 miljoen regels code is rendabel

INTRAFFIC

Conversie PRL, van PASCAL naar C++

ProRail
CORNERSTONE
INTRAFFIC

Samenwerking

ProRail
- Opdrachtgever voor InTraffic
- Sparring partner voor technische issues
- Voert validatie en ketentesten uit

Cornerstone
- Opdrachtnemer van InTraffic
- Ruime ervaring in taalconversies
- Specialist in automatische conversie

InTraffic
- Spoorse kennis en kennis van software ontwikkeling en testen
- Voert handmatige conversie, integratie en systeemtest uit

Conversie PRL, van PASCAL naar C++

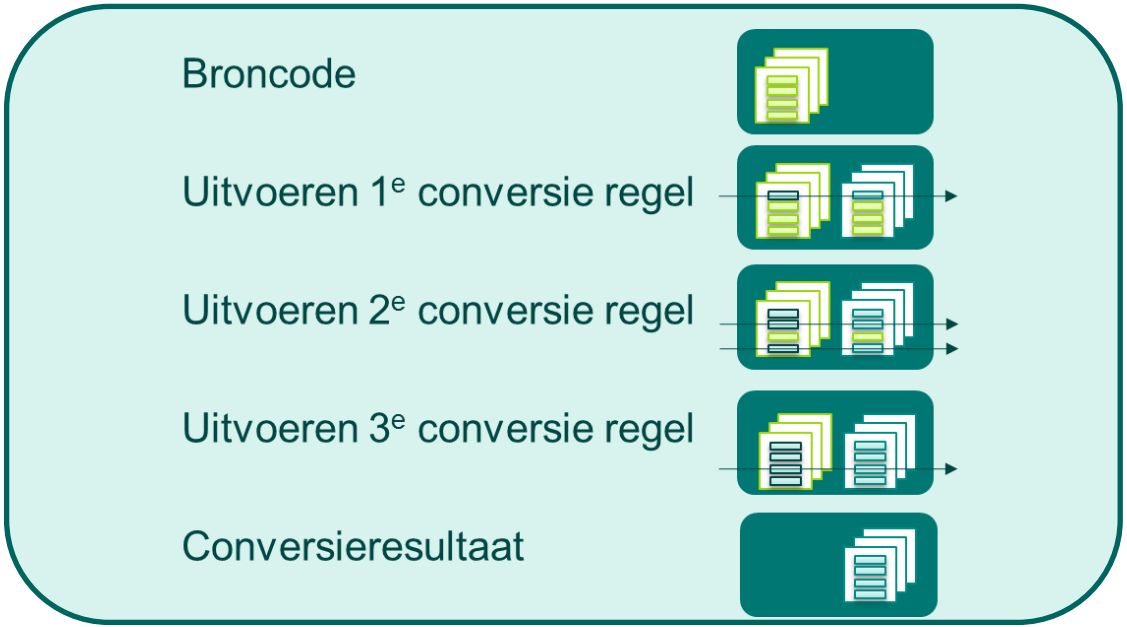
Acceptatie

- Als de geconverteerde PRL functioneel gelijk is aan de Pascal versie van PRL.
- Responstijden en CPU belasting wijken niet meer dan 10% t.o.v. Pascal versie.
- in geen geval zullen er taalconstructies zijn toegepast die latere migratie naar Linux bemoeilijken.

Ontwikkelen van een convertertool

Methode conversie

- Metafoor: Bij het vertalen van een boek wordt niet bladzijde voor bladzijde vertaald maar worden woorden of zinconstructies die in het boek voorkomen in één keer voor het gehele boek vertaald.
- Principe: Een code constructie wordt altijd op dezelfde manier vertaald door de gehele sourcecode heen.

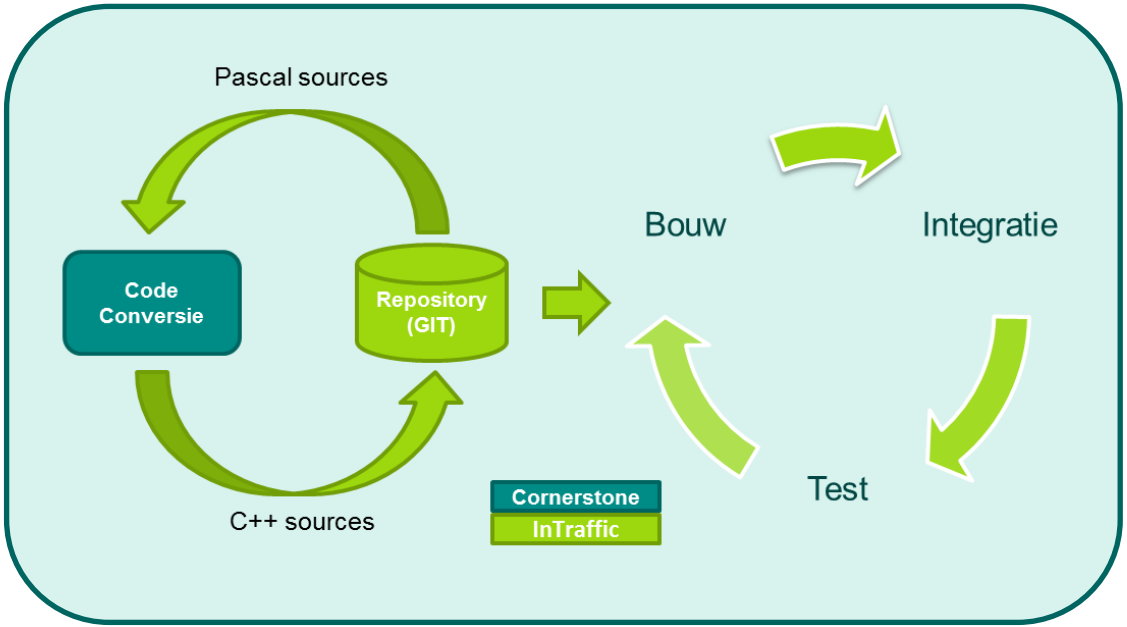


INTRAFFIC

Testen of de geconverteerde code voldoet

Testen: aansluiten op conversie

- Oplevering testbaar conversie-resultaat in clusters
- Bouw, integratie en test

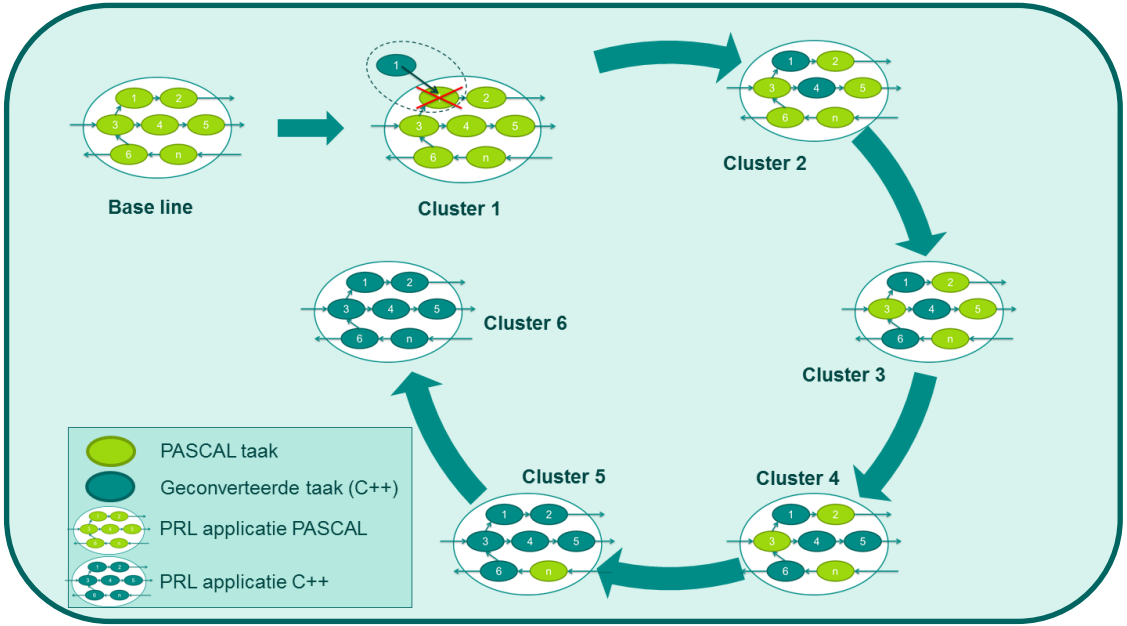


INTRAFFIC

Testen of de geconverteerde code voldoet

Integratie

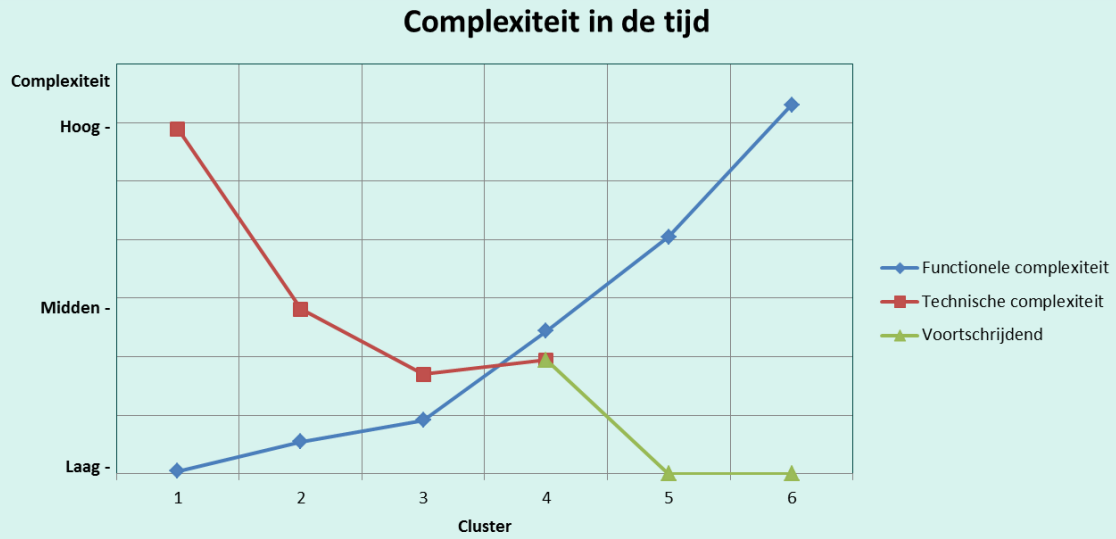
- Hybride testomgeving met altijd de volledige applicatie
- Beheersbaarheid: in stappen integreren door PASCAL taken te vervangen door C++ taken
- Flexibiliteit: geleidelijk opbouw van complexiteit bepaalt de Integratievolgorde



INTRAFFIC

Testen of de geconverteerde code voldoet

Testaanpak



Algemeen

- Automatische conversie: taalconstructie één keer goed geconverteerd betekent alle keren goed geconverteerd

Testen in eerste clusters:

- Complexe taalconstructies
- Functioneel eenvoudige taken

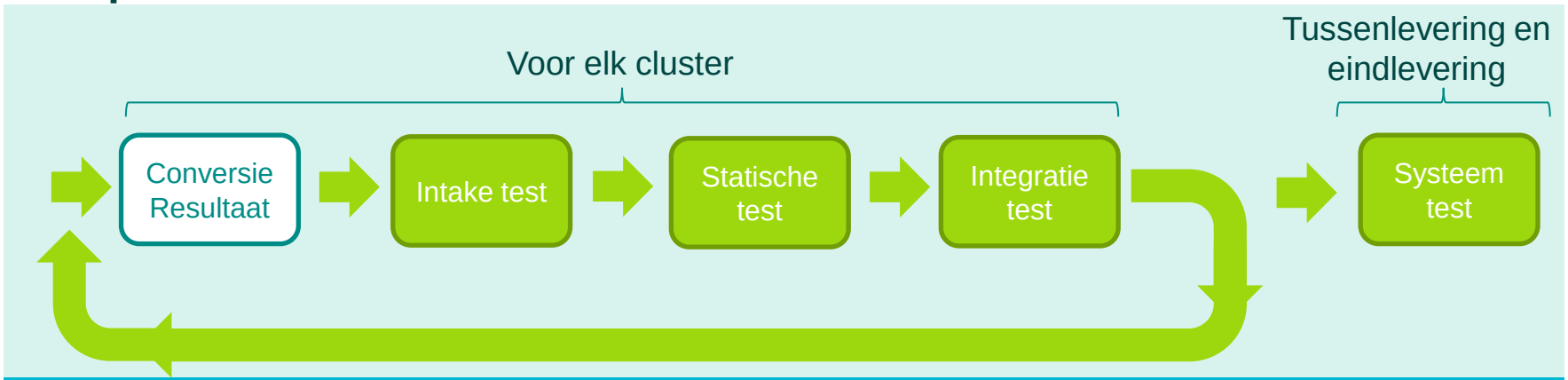
Testen in latere clusters:

- Eenvoudige taalconstructies
- Functioneel complexe taken

INTRAFFIC

Testen of de geconverteerde code voldoet

Testproces



- Intake test: is verder testen zinvol?
- Statische test: controle op onderhoudbaarheid
- Integratie test: dynamische test voor aantonen integrale gedrag
- Systeemtest: voldoen aan de acceptatiecriteria

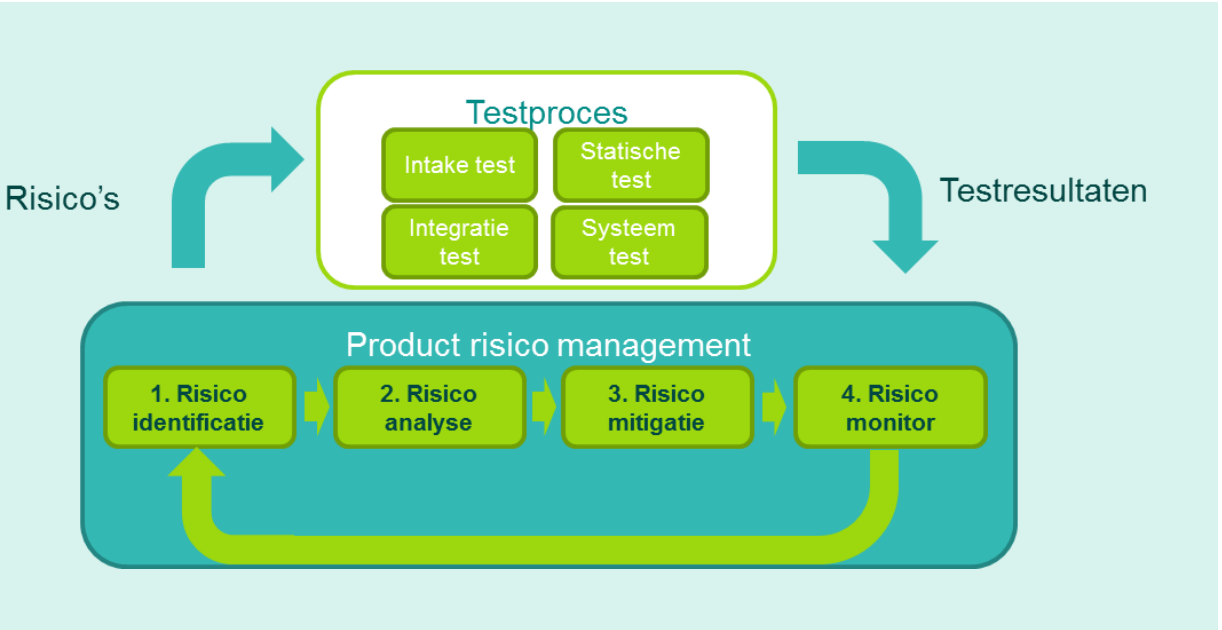
INTRAFFIC

Testen of de geconverteerde code voldoet

Testproces is volledig gestuurd door risico's

Risico's
- Sturen het testproces
- Productrisicomanagement:
  een continue proces
- Proces in 4 stappen
- Productrisico's
  - Indeling volgens FMEA
  - Initieel een PRA
  - Ingedeeld naar kwaliteitskenmerk
  - Gerelateerd aan de testsoorten

Testresultaten
- Meenemen van de opdrachtgever in testresultaten creëert draagvlak en betrokkenheid



INTRAFFIC

Testen of de geconverteerde code voldoet

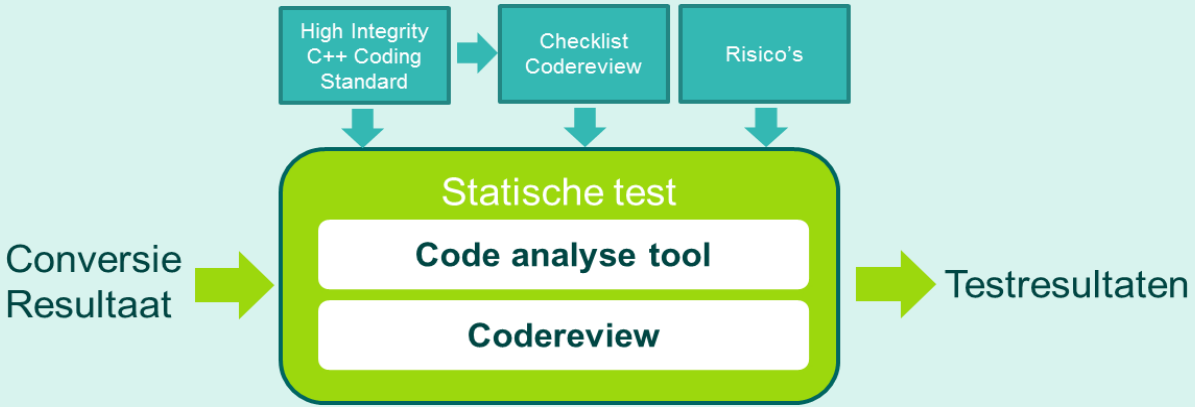
Statische test: controle op onderhoudbaarheid

Risico's
- Gerelateerd aan onderhoudbaarheid

Acceptatie criterium
- Geen taalconstructies die migratie naar Linux bemoeilijken.

Code analyse tool
- Tool Flexelint
- Code standaard: High integrity C++

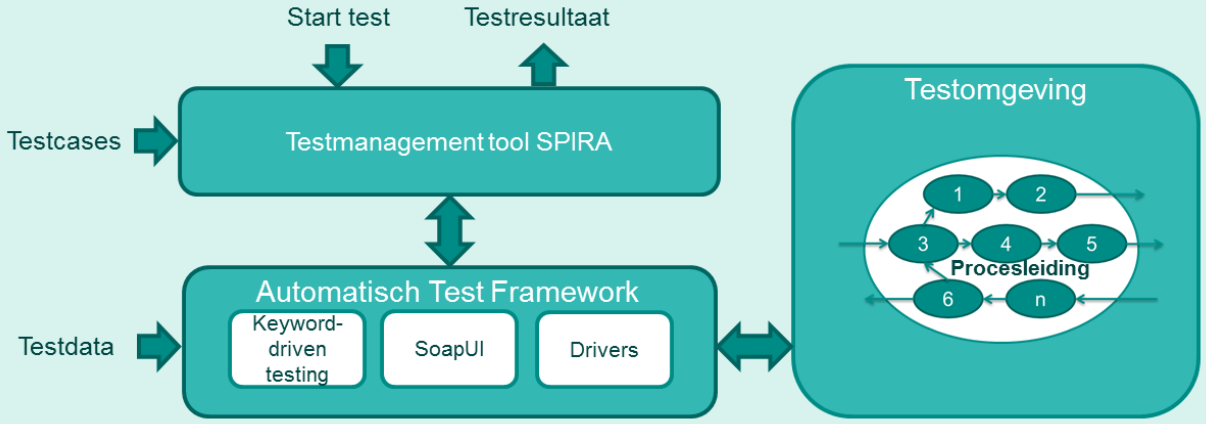
Codereview
- M.b.v. Checklist
  - Leesbaarheid (High integrity C++)



INTRAFFIC

Testen of de geconverteerde code voldoet

Gebruik van testautomatisering en tooling



Automatische Test Framework

- Eigen ontwikkeling: geen geschikte tooling op de markt
- ATF simuleert interfaces en controleert verwachte resultaten

Opbouw

- SPIRA voor opslag van testcases en vastlegging van testresultaten
- SoapUI voor testuitvoering
- Keyword-driven testing
- Drivers

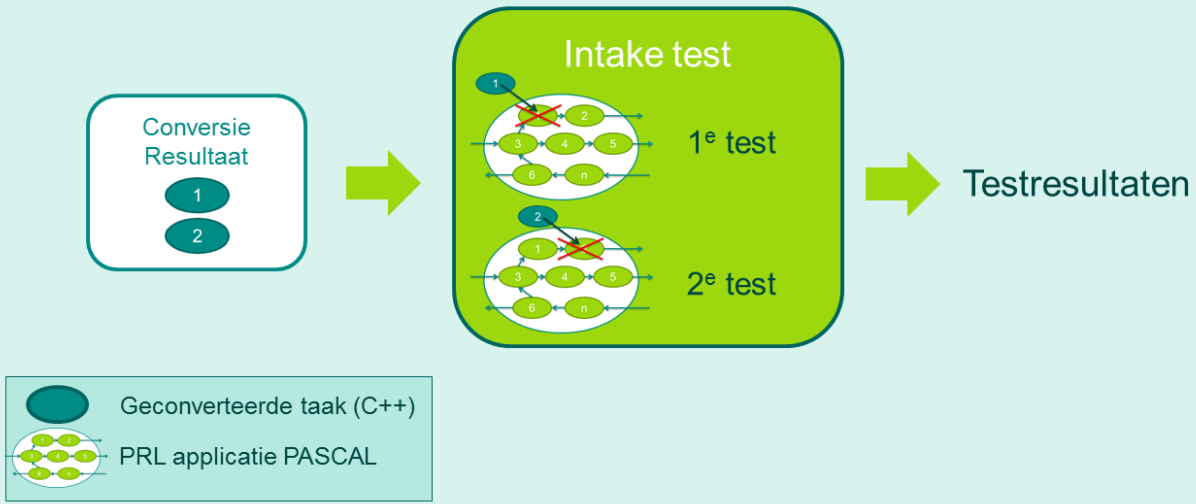
Gebruik

- Behalen van een hoge testdekking
- Reproduceerbaar testresultaat

INTRAFFIC

Testen of de geconverteerde code voldoet

Intake test: is verder testen zinvol?



Geconverteerde library functies

- Automatische unittesten
- Specifiek geschreven code om library functies te testen

Geconverteerde taak

- PRL in PASCAL omgeving
- Automatische Test Framework
- Elke test 1 geconverteerde taak
- Test van een eenvoudig “goed” pad

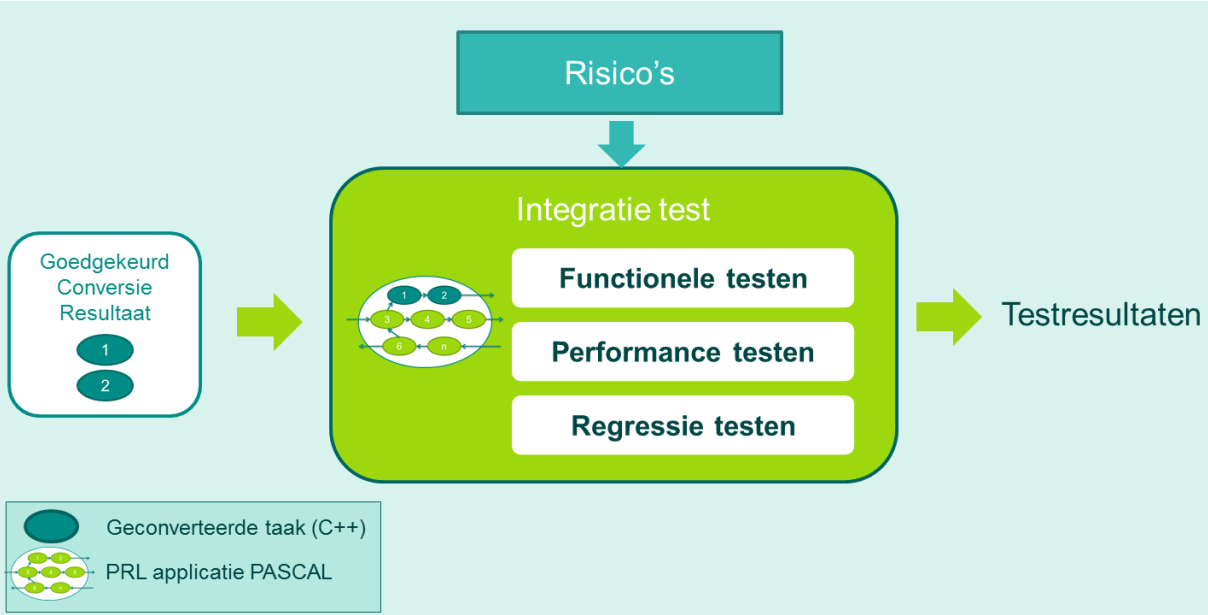
INTRAFFIC

Testen of de geconverteerde code voldoet

Integratie test: dynamische test voor aantonen integrale gedrag

Risico's
- Gerelateerd aan:
  - Functionele correctheid
  - Koppelbaarheid
  - Betrouwbaarheid
  - Performance

Testen
- PRL in PASCAL omgeving + geconverteerde taken
- Automatische Test Framework
- Geen regressie tegenover voorgaande clusters



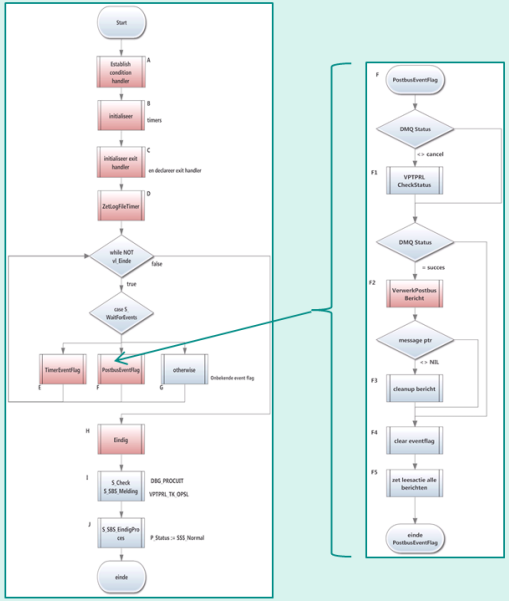
INTRAFFIC

Testen of de geconverteerde code voldoet

Integratie test: Functionele testen

In eerste clusters
- Complexe taalconstructies
- Functioneel eenvoudige taken
- Codepaden in PASCAL
- Analyse tool t.b.v. codepaden
- Statementcoverage

In latere clusters
- Eenvoudige taalconstructies
- Functioneel complexe taken
- Op basis van functionaliteit



Happy flow 1: logbericht, nieuwe logfile file niet open

F1 – F2A – F2E1 – F2E2 – F2E3 – F2E4A – F2E4B – F2E4D – F2E4F – F2E4G – F2E4H – F2E4I – F2E5 – FE2E6 – F2E7 – F2E8 – F2E9 – F2E10 – F2D1 – F2D2 – D3 – F2H – F3 – F4 – F5

Happy flow 2: logbericht, filenaam = openfilenaam

F1 – F2A – F2E1 – F2E2 – F2E3 – F2E4A – F2E4B – F2E4D – F2E5 – FE2E6 – F2E7 – F2E8 – F2E9 – F2E10 – F2D1 – F2D2 – D3 – F2H – F3 – F4 – F5

Alternative flow: logbericht, nieuwe logfile file open

F1 – F2A – F2E1 – F2E2 – F2E3 – F2E4A – F2E4B – F2E4D – F2E4E – F2E4F – F2E4G – F2E4H – F2E4I – F2E5 – FE2E6 – F2E7 – F2E8 – F2E9 – F2E10 – F2D1 – F2D2 – D3 – F2H – F3 – F4 – F5

Alternative flow: logbericht, nieuwe logfile file, openen mislukt 1 keer

F1 – F2A – F2E1 – F2E2 – F2E3 – F2E4A – F2E4B – F2E4D – F2E4E – F2E4F – F2E4G – F2E4J – F2E4K – F2E4H – F2E4I – F2E5 – FE2E6 – F2E7 – F2E8 – F2E9 – F2E10 – F2D1 – F2D2 – D3 – F2H – F3 – F4 – F5

Alternative flow:: Logbericht, bericht niet verwerken

F1 – F2A – F2E1 – F2E2 – F2H – F3 – F4 – F5

Als het bericht verwerkt wordt, dan wordt de naam van het deelsysteem aangevuld met spaties tot 15 karakters en het TBS id tot 12 karakters.

Met input partitioning zijn dan de volgende situaties te onderscheiden voor de stappen F2E7 en F2E8:

	ondergrens	exact	bovengrens
F2E7, deelsysteem	<15	15	>15
F2E8, TBS id	<12	12	>12

INTRAFFIC

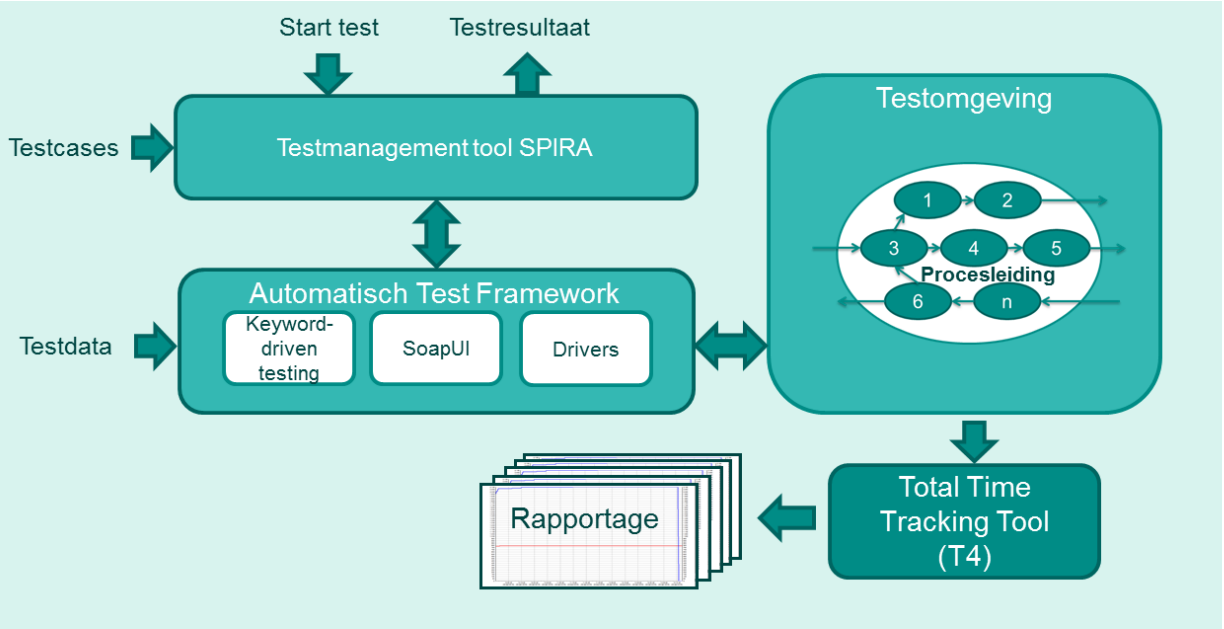
Testen of de geconverteerde code voldoet

Integratie test: Performance testen

Focus
- Vanaf eerste clusters

Meten
- Responstijden
- CPU belasting
- Dynamisch geheugengebruik
- Schijfgebruik
- Verwerkingssnelheid
- Opstartgedrag applicatie

Test
- Tooling: Total Time Tracking Tool
- Resultaat relateren aan PASCAL



INTRAFFIC

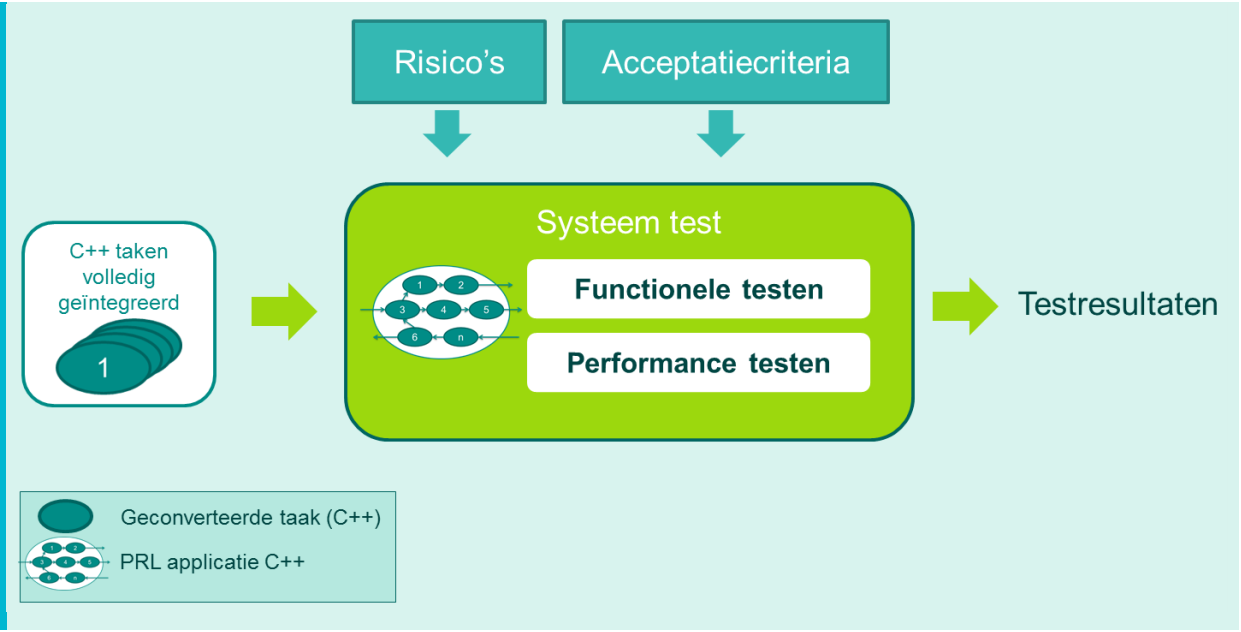
Testen of de geconverteerde code voldoet

Systeem test: voldoen aan de acceptatiecriteria

Randvoorwaarden
- Applicatie is volledig geïntegreerd
- “Representatieve” testomgeving

Risico’s
- Gerelateerd aan:
  - Functionele correctheid
  - Koppelbaarheid
  - Performance
  - Betrouwbaarheid

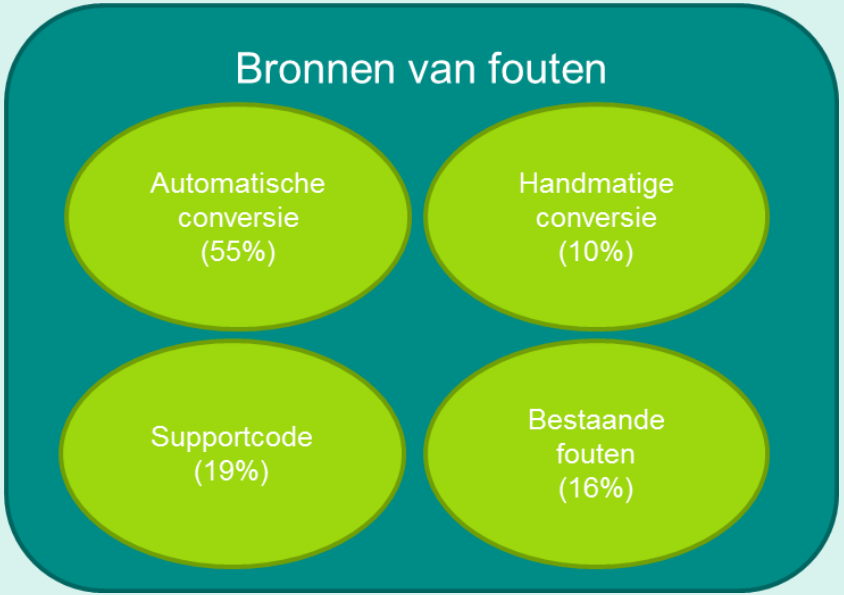
Acceptatiecriteria
- betreft functioneel gedrag
- betreft responsetijden en CPU belasting



INTRAFFIC

Testen of de geconverteerde code voldoet

Gevonden fouten



INTRAFFIC

Conclusie

- Zoek samenwerking met een partij, gespecialiseerd in automatische code conversie.
- Bij automatische conversie hoeft niet alle code getest te worden: een taalconstructie één keer goed geconverteerd betekent alle keren goed geconverteerd.
- Start codeconversie met een functioneel eenvoudige taak en test deze grondig.
- Continue monitoren van productrisico's op basis van testresultaten om het testproces bij te sturen.
- Gebruik een testomgeving waar het mogelijk is om bestaande applicatiedelen en geconverteerde applicatiedelen met elkaar te laten samenwerken.
- Met geautomatiseerd testen kun je in korte tijd vaststellen of aan het acceptatiecriterium van functioneel gelijk gedrag is voldaan met hoge testdekking en gegarandeerde reproduceerbaarheid.

Vragen

INTRAFFIC