

NR.15 MEI 2011

/Java; what's next?

\*technologie

/Falcon elektronendetectiecamera

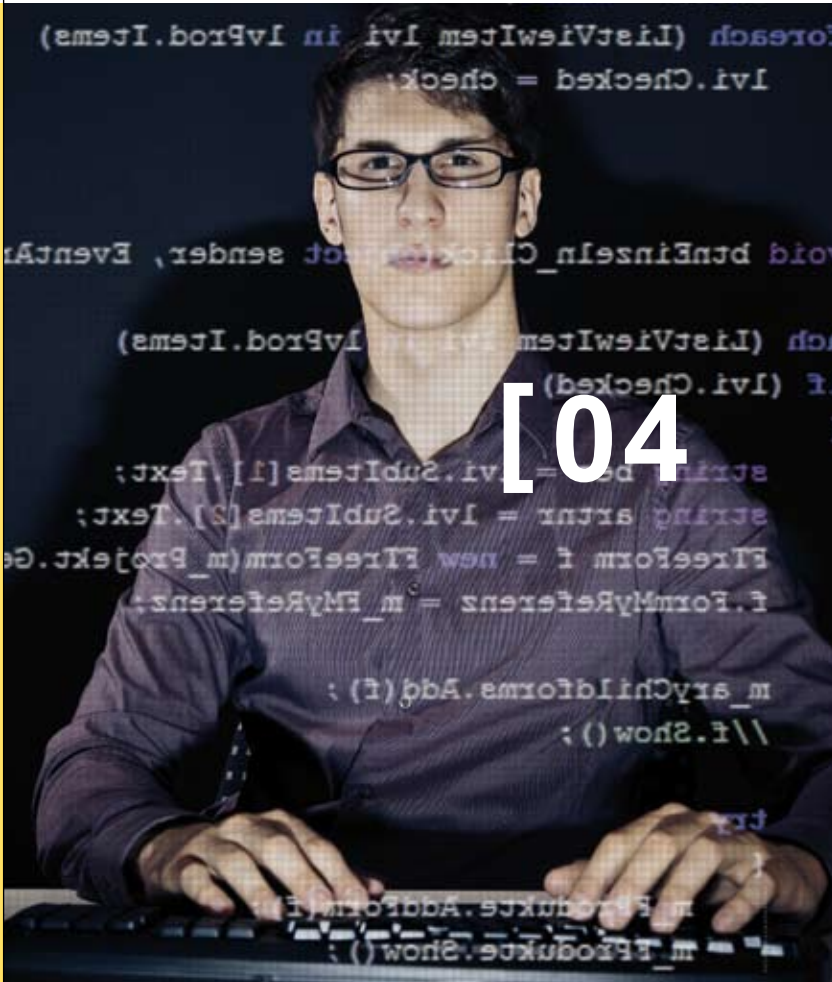
\*case

/smart grids

\*visie

/softwarekeurmerk

\*trends & hypes



## /inhoud

### [03] /voorwoord

#### \*de ultieme energiebron bestaat niet

We zullen bronnen moeten combineren om te kunnen blijven voorzien in onze energiebehoefte.

### [04] /technologie

#### \*Java: what's next?

De wondere wereld van programmeertalen.

### [08] /visie

#### \*van gebruiker naar energieproducent

Technische innovaties zorgen voor verandering in energiemarkt.

### [10] /case

#### \*zoektocht naar het beste beeld

De Falcon elektronendetectiecamera.

### [12] /interview klant

#### \*Enexis

Alles voor een betrouwbaar energienetwerk.

### [14] /trends & hypes

#### \*softwarekeurmerken: een trend of hype?

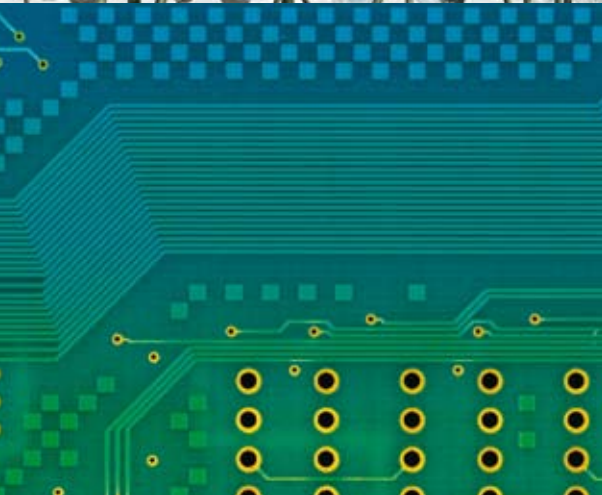
Naast verificatie ook validatie noodzakelijk.

### [16] /medewerker aan het woord

#### \*in gesprek met Enno Romkema



Jan van der Wel  
Algemeen directeur



## /voorwoord

**De olieprijs blijft maar stijgen en kan wel naar 250 dollar per vat oplopen.**

**Het gevolg van de omvallende dictatoriale regimes in diverse Arabische landen, zo wordt gesteld. Of is er wat anders aan de hand? De voorraad fossiele brandstoffen zoals aardolie, aardgas en steenkolen is eindig. Deze brandstoffen kunnen steeds moeilijker gewonnen worden, daarnaast eisen opkomende economieën een steeds groter deel van deze voorraden op.**

Alternatieve energiebronnen zijn een oplossing, maar ieder alternatief heeft zijn eigen beperkingen. Zo zijn er biobrandstoffen die ten koste van onze voedselvoorziening gaan. Windenergie hebben we alleen als er wind is en zonne-energie alleen overdag. Kernenergie kent grote risico's, zoals we helaas recent met eigen ogen hebben kunnen zien in Japan. De ultieme energiebron bestaat dus niet en we zullen bronnen moeten combineren om te kunnen blijven voorzien in onze steeds groter wordende energiebehoefte.

## >>de ultieme energiebron bestaat niet

Een mooi voorbeeld van de ontwikkeling op dit gebied is de elektrische auto. Onze auto's rijden nu nog voor bijna 100% op van aardolie afgeleide producten. Een brandstof die uitgeput zal raken. Gaan we in de toekomst rijden op brandstofcellen die met waterstofrijke grondstoffen elektriciteit opwekken of direct op elektriciteit vanuit accu's die opgeladen zijn door het elektriciteitsnetwerk thuis of onderweg? De auto gaat uiteindelijk aangedreven worden door een elektromotor. Welke energiebron deze heeft is nog niet precies duidelijk. Maar wat wel duidelijk is, is dat elektriciteit de uiteindelijke geüniformeerde energiedrager wordt.

Een nadeel aan elektriciteit is dat deze zelf niet opgeslagen kan worden. Hierdoor wordt het voor leveranciers steeds belangrijker om vraag en aanbod goed op elkaar af te stemmen. Want als wij steeds meer elektriciteit verbruiken, moet dat wel beschikbaar zijn. Energiebedrijven en netbeheerders onderzoeken daarom de mogelijkheden van energieopslag en energiemanagement. In deze Objective gaan we in op technologische energie-innovaties, waarbij netbeheerder Enexis haar aanpak vertelt en wij onze visie op slimme netwerken geven.

**Veel leesplezier toegewenst!**

/technologie

# Java; what's next?

## >>de wonderse wereld van programmeertalen

**Een programmeertaal is het gereedschap van de programmeur. In programmeertalen is een eeuwige evolutie gaande. Het dient continu te verbeteren zodat een programma steeds sneller en efficiënter geschreven kan worden en fouten vooraf gedetecteerd worden. Maar is elke nieuwe taal ook een verbetering? En wanneer stap je over op een nieuwe taal?**

Er is een ijzeren wet in computerland: ieder jaar worden processors aanzienlijk sneller. Snellere computers maken het mogelijk dat software steeds meer voor ons kan doen. Programmeertalen vormen het gereedschap om software te bouwen. En ook daar vindt continu evolutie plaats. Er komen regelmatig nieuwe talen bij, waardoor de programmeur efficiënter kan werken. Nieuwe talen komen niet 'uit het niets'; ze bevatten vaak nieuwe elementen, maar ook altijd beproefde elementen uit het verleden.

Als talen een beetje op elkaar lijken, is het leerproces eenvoudiger. Maar als ze heel veel op elkaar lijken, maak je geen vooruitgang. Voor de gebruiker van deze talen noopt dat tot strategische keuzes: blijven we de huidige taal gebruiken of stappen we over op een nieuwe taal? De nieuwe taal heeft de belofte van efficiënter werken, maar vergt wel de inspanning om de nieuwe taal te leren en de hoop dat de rest van de wereld ook volgt.

### Evolutie

Verandering in programmeertalen is van alle tijden. De eerste talen waren lineair. In de eerste 'Basic-dialecten' programmeerde je regel voor regel, met loops en sprongen naar een ander stuk van je code. Vervolgens ontstonden vriendelijkere talen zoals Pascal en C. Dit soort talen noemt men imperatieve of procedurele talen. Binnen de academische wereld is er ook een andere stroming: de functionele talen zoals Lisp en Haskell. In dit soort talen zijn functies het centrale concept. In de industrie zijn echter de functionele talen nooit echt populair geweest.

Een grote innovatie was de stap van imperatieve talen naar object-georiënteerde (OO) talen. De taal Smalltalk is de moeder van de OO-talen, maar met C++ werd deze stroming pas gemeengoed. Java is een OO-taal die nu veel gebruikt wordt, maar eigenlijk

Functional Programming	Haskell / Clojure / F#	
	Scala	Ruby Python
	C#	
Object Oriented Programming	Java	Smalltalk
	Statically Typed	Dynamically Typed

```

switch(Label)
{
    case eNodeLabel.Stage:
        g.FillRectangle(BrushGray, new Rectangle(RectP
        string strperc = Mail * 100.05
        g.DrawString(strperc, MainFont, BrushBlack, R
        break;
    case eNodeLabel.Amount:
        g.FillRectangle(BrushGray, Rect
        string stram = Bau.Kombiniert.ToSt

        foreach(int hyb in Bau.HybridAnzahl)
            stram += (" "+hyb.ToString());

        g.DrawString(stram, MainFont, BrushBlack, R
        break;
    case eNodeLabel.None:
    default:
        g.FillRectangle(BrushGray, RectPercent);
        break;
}

```

moeten we het hebben over het Java-platform. Dit is een drie-eenheid van de taal Java, de Java Virtuele Machine en de Java-bibliotheken.

### De Java-taal

Java stamt uit de jaren negentig en toen was hardware performance een issue. Om toch het onderste uit de kan te halen qua performance en tegelijk de overstap naar die (destijds nieuwe) taal niet te groot te maken, zijn soms concessies gedaan aan de taal. Java is daarom geen zuivere objectgeoriënteerde taal zoals Smalltalk: niet alles is een object. Voor een programmeertaal betekent dit tweeslachtigheid en omgaan met uitzonderingen.

### De Java Virtuele Machine (JVM)

Het toepassen van een virtuele machine is één van de belangrijkste innovaties van het Java-platform. Bij het compileren van C wordt programmacode gegenereerd in machinetaal voor een specifieke processor. Uitvoering door een andere processor vereist dat het C-programma opnieuw gecompileerd wordt voor die andere processor. Bij compileren van Java wordt het programma in bytecode gegenereerd. Bytecode wordt niet direct door een processor uitgevoerd maar door een softwarelaag: de JVM. De JVM is typisch in C geschreven en wel voor een specifieke processor gecompileerd. Dit betekent dat, mits een JVM beschikbaar is, een Java-programma overal kan worden uitgevoerd (write once, run anywhere). Daarnaast doet de JVM zich voor als een ideale processor en vereenvoudigt het programmeerwerk. Belangrijk daarbij is het eenvoudige geheugenbeheer en de eliminatie van vele soorten fouten door strenge controles bij compileren en uitvoeren.



## Scala:

```
case class Clock (var hour:
Int, var min: Int);
```

## Java:

```
public class Clock {
    private int hour;
    private int min;

    public Clock(int hour,
int min) {
        this.hour = hour;
        this.min = min;
    }

    public int getHour() {
        return hour;
    }

    public void setHour(int
hour) {
        this.hour = hour;
    }

    public int getMin() {
        return min;
    }

    public void setMin(int
min) {
        this.min = min;
    }

    public String toString() {
        ...
    }

    public boolean
equals(Object arg1) {
```

## De Java-bibliotheken

Voor de productiviteit is het ook van belang dat er genoeg bibliotheken beschikbaar zijn. In een bibliotheek staat code die al door anderen is geschreven. Een populaire taal kan megabytes aan reeds geschreven code in bibliotheken hebben, waar je maar een paar regels aan toevoegt om bijvoorbeeld een website te maken. De bibliotheek is dus ook een overweging bij de keuze voor een taal. Java bestaat al ruim tien jaar en in die tijd zijn er heel veel bibliotheken ontwikkeld voor Java. Veelal is van deze bibliotheken ook de Java-code vrij (open source) beschikbaar.

## >>iedere taal heeft zijn eigen levenscyclus

### De ontwikkelomgeving

Naast het Java-platform is voor de productiviteit ook nog de ontwikkelomgeving van groot belang. Deze ontwikkelomgevingen worden ook wel 'Integrated Development Environment' (IDE) genoemd. Voor Java zijn er een aantal beschikbaar en Eclipse is hiervan het meest bekend. De IDE geeft de programmeur veel ondersteuning bij het programmeren; zo controleert het al tijdens het invoeren of de code correct is. Het bevat helpfuncties, ingebouwde debuggers en links naar bibliotheken.

### Wet van de remmende voorsprong

De innovatie in een taal is omgekeerd evenredig met de leeftijd van de taal en het aantal gebruikers. Softwareproducten van een taal die al in gebruik zijn, noodzakelijk tot 'backwards compatibility'. Een aspect dat voormalig Java-eigenaar Sun altijd heel goed in de gaten heeft gehouden. Maar dat remt de innovatie tot het punt dat de taal stilstaat. Dan is het tijd om uit te kijken naar een nieuwe taal. Iedere taal heeft dus zijn levenscyclus.

### What's next?

De laatste jaren duiken er steeds nieuwe talen op. Een aantal talen bouwt voort op wat er al was en maakt gebruik van de JVM met alle voordelen van dien. Voorbeelden zijn Ruby, Python en Groovy, die over het algemeen veel compactere code geven dan Java. Ook zijn innovaties in deze talen doorgevoerd zoals 'closures' waar een functie eenvoudig wordt uitgeschreven op de plaats waar die nodig is. Voor kleine functies bespaart dit veel typewerk. De Java-gemeenschap is al jaren aan het vergaderen over hoe ze dit alsnog in Java kunnen inbouwen.

Een belangrijk nadeel van de hierboven genoemde nieuwe talen is dat ze allemaal gebaseerd zijn op 'dynamic typing'. Dit betekent dat je fouten pas vindt bij het uitvoeren van het programma. Voor het bouwen van complexe programma's met grote teams zijn deze talen daarom niet geschikt. Je wilt juist dat de compiler je maximaal helpt met het vinden van mogelijke fouten in het programma, nog voordat het wordt uitgevoerd.

## Scala

Een positieve uitzondering is de nieuwe taal Scala. De persoon achter deze taal is Martin Odersky, tegenwoordig professor aan de universiteit van Lausanne (EPFL) en een belangrijk persoon achter één van de laatste innovaties in Java: Generics. Een Scala-programma wordt uitgevoerd op de JVM en kan gebruik maken van de bestaande Java-bibliotheken. Scala is een taal met relatief veel innovatie. Het is een hybride taal waarin OO-concepten en functionele concepten samenkomen. Deze gedachtegang staat niet op zichzelf, maar is een trend. Ook Microsoft maakt hiervan gebruik in het .Net-platform door middel van LINQ. Zonder hier verder op in te gaan kan van LINQ gezegd worden dat het zwaar leunt op het gedachtegoed van de functionele talen.

Verder is belangrijk dat Scala gebruik maakt van 'static typing'. Dit betekent dat de compiler streng is en helpt in het voorkomen van vele soorten fouten. Een opvallende innovatie is 'type inference'. Dit betekent dat een programmeur niet altijd expliciet de types van variabelen en parameters hoeft te herhalen, met compactere code als resultaat. Eigenlijk bepaalt de compiler, indien mogelijk, zelf de types. Het wetenschappelijke idee achter het beredeneren van de types door de compiler komt ook weer uit de functionele wereld. Een andere innovatie zijn 'traits'. Dit zijn Java-interfaces waarin ook implementatie (velden en code) is toegestaan. Met traits kunnen lastige beperkingen van enkelvoudige overerving overwonnen worden.

De Scala-gemeenschap groeit snel en heeft een IDE ontwikkeld voor Eclipse die al redelijk werkt. Scala heeft hiermee alle ingrediënten in zich als potentieel opvolger van Java... toch?

## Drempel

Het speelveld overziend heeft Scala inderdaad goede papieren, maar zo hard gaat het natuurlijk ook weer niet. De taaldrempel is voor de meeste gebruikers nu nog te hoog, omdat door de vele innovaties de taal Scala toch behoorlijk verschilt van de taal Java. Dat zien we vaker bij de overgang naar een nieuwe taal. Eerst zul je moeten investeren, voor je de vruchten van de verbeterde productiviteit kunt plukken. Ook werkt de IDE wel, maar is nog niet zo handig als dat men bij Java gewend is.

Daarnaast lijkt de Java-gemeenschap een beetje zijn kop in het zand te steken en liever nog een jaartje te vergaderen over een nieuwe verbetering aan de taal. Hier spelen wellicht ook politieke zaken die niets te maken hebben met de innovaties in de taal.

Zolang grote bedrijven een nieuwe taal niet oppikken, zal Scala een nicheproduct blijven. Een bedrijf als Google zou Scala het benodigde gewicht kunnen geven om over het kantelpunt heen te komen. De zoekgigant heeft ook vaak het lef getoond om iets nieuws uit te proberen. Gelukkig zie je al wel dat Scala kan meedraaien in het 'Summer of code'-programma van Google, waar het sponsor is van universitair vakantiewerk voor veelbelovende projecten.

Voor Technolution is belangrijk dat we het speelveld van de programmeertalen blijven volgen, dat we daarbij het kaf van het koren scheiden en dat we op tijd in de juiste trein stappen. Eén ding is echter zeker: er komt een dag dat Java weer gewoon een eiland in de Indische Oceaan is.



/visie

# Van gebruiker naar energieproducent

>>technische innovaties zorgen voor verandering in energiemarkt

**Het elektriciteitsnet van de toekomst kent vele gezichten. Er wordt volop gediscussieerd over decentrale opwekking en elektrisch rijden is veel in het nieuws. Om dit mogelijk te maken is een slim energienetwerk nodig; een smart grid. De energiemarkt is dus in ontwikkeling. Er gebeurt veel in korte tijd, waar brengen deze ontwikkelingen ons heen?**

Elektriciteit speelt een grote rol in onze samenleving en dat zal alleen maar toenemen, bijvoorbeeld in mobiliteit. Fossiele brandstoffen zijn eindig, dus we zullen op termijn naar voertuigen over moeten die op een andere manier aangedreven worden, bijvoorbeeld door een motor op elektriciteit. Hiervoor moet het elektriciteitsnet wel gereed gemaakt worden, zodat het niet overbelast raakt als iedereen 's avonds thuis komt en tegelijk zijn accu gaat opladen. Maar die autoaccu's vormen samen ook een enorme opslagcapaciteit. Een buffer om tijdelijke overschotten in op te slaan. Daar moet het elektriciteitsnet wel mee om kunnen gaan. Immers, stroom volgt de weg van de minste weerstand. Vanuit de centrale druppelt de stroom via allerlei afsplitsingen omlaag naar de gebruiker: eenrichtingsverkeer dus. Een slim energienet stuurt de stroom de juiste kant op: van centrale naar stopcontact, maar ook van het zonnepaneel thuis naar een wijkstation of de accu van je elektrische auto. En als er te weinig elektriciteitsaanbod is of er dreigt een overbelasting op het net, draait het huis even op stroom uit accu's van elektrische auto's of uit opgewekte en opgeslagen energie op wijkniveau. Deze accu's worden zo onderdeel van het elektriciteitsnet.

## Smart grid is voor iedereen anders

Dit is slechts één van de vele visies op het 'smart grid', zoals het elektriciteitsnet van de toekomst is gedoopt. Wat een smart grid

precies inhoudt, kan niemand nog vertellen. Iedereen heeft zijn eigen beeld. De netwerkbeheerder ziet het smart grid als een middel voor betere benutting van het netwerk. Dit voorkomt kostbare investeringen als het aanleggen van extra kabels. Het leveringsbedrijf produceert stroom en verkoopt dat aan de klant. Voor hem biedt het smart grid extra mogelijkheden om energie te verkopen aan de klant, bijvoorbeeld via laadpalen voor auto's. En de afnemers van energie worden steeds vaker ook leverancier dankzij zonnepanelen, HRe-ketels, etc. Die hebben weer hun eigen visie op het smart grid.

## Energiemanagement

De beheerders van het elektriciteitsnet moeten inspelen op de steeds grotere schommelingen in vraag en aanbod. De komst van elektrische auto's, warmtekrachtcentrales, accu's, zonnepanelen en CV-ketels die ook stroom leveren (HRe-ketels) zullen het elektriciteitsnet en het management ervan drastisch veranderen. De stroom gaat alle kanten op. En met wind- en zonne-energie varieert het aanbod sterk. Bij een te groot aanbod raken de knooppunten in het net overbelast. Het eerste knooppunt dat daar last van krijgt, is de wijktransformator. Een accu kan het wijkstation ontlasten.

Precies dat gebeurt nu met een proef in Etten-Leur, waar tachtig huizen met zonnepanelen zo'n 200 kWh per dag produceren. Daar komt een enorme accu bij en meet- en regeltechniek om te zien of er inderdaad overschotten zijn. De accu vormt een buffer naar twee kanten. Het slaat overschot van de zonnepanelen op. Maar als de vraag in de wijk hoger is dan de netbeheerder kan leveren op piekmomenten, dan springt de accu bij om het tekort aan te vullen. Of de bewoner stelt zijn slimme wasmachine zo in dat die pas gaat draaien als er genoeg stroom uit zijn eigen zonnepanelen komt. Etten-Leur krijgt zo een smart grid op wijkniveau.

### Verkeersmanagement op het elektriciteitsnet

Een groep wasmachines of vaatwassers die je afhankelijk van de belastbaarheid van het elektriciteitsnet gecontroleerd aanzet, dat klinkt al bijna als toeritdosering op de snelweg. Zo bekeken vertoont het elektriciteitsnet overeenkomsten met het wegennet en is het management van het elektriciteitsnet op te vatten als verkeersmanagement: zorgen dat de vervoersvraag en -capaciteit in balans blijven.

Een belangrijk punt is of de vraag naar stroom kan worden voorspeld. Netbeheerders gebruiken nu al modellen die de stroomconsumptie per etmaal inschatten. Maar in een smart grid, met decentrale opwekking en grote stroomslurpers als elektrische auto's, is een veel nauwkeurigere voorspelling nodig. Zeker als de aantallen elektrische auto's grote vormen gaan aannemen. De meet- en regelapparatuur daarvoor ontbreekt nu nog. Die zal er komen in de vorm van een informatienetwerk rondom het elektriciteitsnet. Energie wordt met ICT verweven om op elke gebeurtenis te kunnen anticiperen.

### Internet van energie

Daar komt een andere analogie om de hoek: het internet. Een overeenkomst die Europa ook ziet, getuige het EU-onderzoeksprogramma 'Internet of Energy', waarin Technolution deelneemt. Hierbij wordt gekeken of het Europese energienetwerk opgezet kan worden zoals het internet. Een flexibel netwerk met links tussen energieproducenten, -leveranciers en -verbruikers waarbij informatie over verbruik, voorraad, opbrengsten en kosten wordt uitgewisseld binnen een veilig netwerk. Een internet van energie vraagt ook aanpassingen in de apparatuur. Die moet extra's hebben om op afstand uitleesbaar en bestuurbaar te zijn. In het EU-project probeert het consortium onder andere tot standaarden te komen voor de koppeling met elektrische apparaten in huis. Deze koppelingen meten het actuele verbruik of het aantal kWh dat via zonnepanelen wordt teruggeleverd aan het net. De consument krijgt zo inzicht in het afzonderlijk verbruik of de opbrengst van elk apparaat. Dat maakt mensen bewust van hun gedrag en het effect daarvan op het milieu en hun portemonnee. Ook dat is het smart grid.

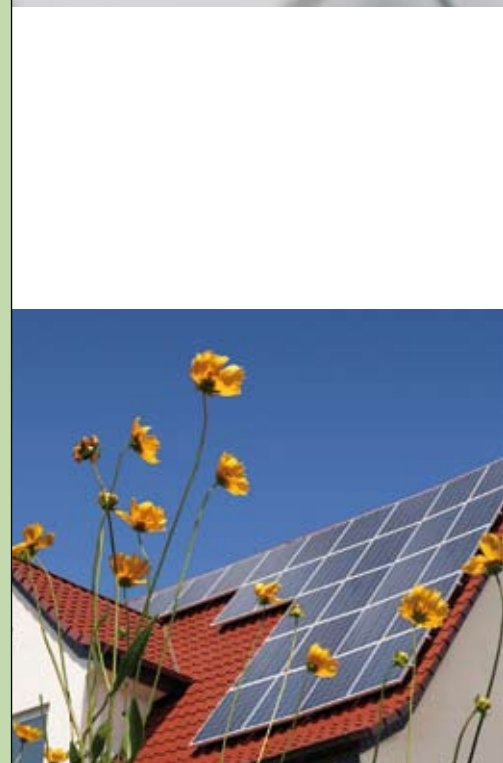
### Rol overheid

De overheid kan met subsidies bewegingen in gang zetten. Zoals ze nu doet met de stichting 'E-laad', waarin netbeheerders samen een netwerk van laadpalen voor elektrische auto's neerzetten. De EU subsidieert onderzoeksprojecten om een Europese markt te creëren. Zo kan een Europese standaard ontstaan, zodat Europese bedrijven zelf de kennis ontwikkelen en niet afhankelijk zijn van andere standaarden.

Maar nog veel belangrijker is wet- en regelgeving. De overheid kan bij het verlenen van concessies aan streekvervoer en groepsvervoer eisen stellen op milieugebied. Gemeentes kunnen de logistieke sector stimuleren door in stedelijke milieuzones te verplichten de winkelbevoorrading met elektrische wagens te doen. Dat scheelt een boel herrie en stank. Bedrijven spelen al in op deze trends en werken aan elektrische vrachtwagens en bussen die inductief opladen bij halteplaatsen en op het eindpunt.

### Eerste levensfase smart grid

Hoe het smart grid er precies gaat uitzien, is dus nog onduidelijk. Wel zeker is dat het drie markten gaat samenbinden: energie, verkeer & vervoer en financiën. Markten waarin Technolution al actief is en uitgebreide domeinkennis van heeft. Het smart grid zit nu in de fase van ontdekken en ontginnen. Standaarden bestaan nog niet maar zijn wel in de maak. Dit is de fase van prototypes bouwen en concepten testen, en anticiperen in de zoektocht naar de juiste richting voor het smart grid.



/case

# Zoektocht naar het beste beeld

## >>de Falcon elektronendetectiecamera

**Biologische materialen zijn lastig te onderzoeken in een elektronenmicroscop. Het materiaal heeft van nature weinig contrast en kan onder invloed van elektronen degraderen. Met een nieuw type sensor kan een lagere dosis gebruikt worden waardoor meer informatie verkregen kan worden voordat het monster gedegradeerd is. Technolution heeft op basis van deze sensor een camera ontwikkeld voor microscoopbouwer FEI.**

FEI is wereldleider in de fabricage van high-end elektronen-microscopen. Met zogenaamde transmissie elektronenmicroscopen (TEM) kunnen details van minder dan een Ångstrom (10-10 m) worden onderscheiden. Gebruikers van deze microscopen zijn te vinden in de elektronica-industrie, in de biomedische wereld, bij materiaalonderzoek en in een brede range aan onderzoeksinstituten.

In een TEM gaan de elektronen door het preparaat heen, zoals het licht in een gewone (licht) microscoop of diaprojector. Elektromagnetische lenzen vergroten de bundel en beelden het preparaat af op een camera, die de beelden naar een pc stuurt. Biologische materialen hebben weinig contrast in een TEM en vallen snel uit elkaar door de elektronenstraling. Voor biologisch onderzoek is een lage dosis elektronen wenselijk, maar met een conventionele camera gaat dat gepaard met een slechte signaal-ruisverhouding en een lage resolutie. Daarom heeft FEI voor deze toepassing een gevoeliger CMOS-sensor laten ontwikkelen voor de directe detectie van elektronen.

### **Van sensor naar directe detectie**

In conventionele camera's voor TEM worden de elektronen met een scintillator omgezet in licht, dat vervolgens wordt geregistreerd door de sensor. Dit conversieproces is nodig om de sensor te beschermen tegen de elektronen, maar gaat gepaard met een verlies van efficiëntie en resolutie. De nieuwe sensor is speciaal ontworpen om bestand te zijn tegen de invloed van directe belichting met elektronen en is daardoor veel gevoeliger. Het resultaat is dat vergeleken met een gewone sensor bij een gelijke dosis

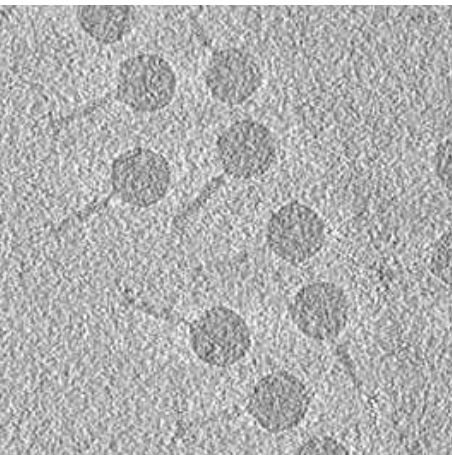
drie keer zoveel beelden kunnen worden opgenomen met dezelfde kwaliteit. Daarnaast is voor een enkel beeld maar een derde van de dosis nodig.

### **Van sensor naar camera**

Een sensor is nog geen camera. De essentie om van een sensor een camera te maken is signaalbewerking. Maar er moet natuurlijk ook een behuizing omheen. De sensor is een forse siliciumchip van 16 megapixels. De chip komt op een print en dit geheel moet in de bestaande sensorkamer van de TEM passen. De beschikbare fysieke ruimte voor elektronica is daardoor zeer beperkt. Door de sensor direct op de print te wirebonden is deze ruimte efficiënt benut. Tevens stelt het hoog vacuüm in de sensorkamer voorwaarden aan het printplaatontwerp en het gebruikte materiaal om vervuiling van het vacuüm tegen te gaan. Verder moet het mogelijk zijn de sensor terug te trekken uit de sensorkamer om zo ruimte te maken voor een tweede camera; dit vereist speciale flexibele printplaattechnieken. Natuurlijk worden er ook hoge eisen gesteld aan de signaaloverdracht van de sensor output. Om aan deze eisen te voldoen is gebruik gemaakt van simulaties en field solvingtechnieken, waarbij het gedrag van het analoge signaalpad goed geanalyseerd en geoptimaliseerd is.

### **Vacuüm & koeling**

Zoals eerder gezegd werkt een elektronenmicroscop in hoog vacuüm, dit brengt extra technische uitdagingen met zich mee. De elektronenmicroscop werkt in hoog vacuüm om verstrooiing van de elektronenbundel te voorkomen. De sensor bevindt zich ook in het vacuüm, waar de mogelijkheden voor koeling beperkt zijn.



Koeling is nodig voor een betere signaal-ruisverhouding. De sensor wordt met een Peltier-element gekoeld en de warmte wordt afgevoerd via waterkoeling.

Vanwege de beperkte koeling en ruimte wordt de benodigde elektronica zoveel mogelijk buiten het vacuüm gehouden. Op een vacuüm compatible print wordt de camera uitgelezen. De ruwe beelddata wordt direct door de vacuümbehuizing gevoerd, gedigitaliseerd en via een 10Gb-glasvezellink weggeleid van de camera. De link gaat naar een externe processingunit, waar de beeldbewerking plaatsvindt.

### Beeldbewerking

Elke pixel wordt apart gekalibreerd voor offset en gain. Op zo'n grote chip (4K bij 4K, dus 16 megapixels) heb je grote kans op defecte pixels. Ook dat kan de software compenseren.

Aansturing van de algoritmes gebeurt op commando van de TEM-software die op de pc van de operator draait. Uiteindelijk beslist de operator welke knoppen hij aan of uit zet. Door de grote hoeveelheid aan pixeldata is het niet mogelijk om deze algoritmes met een standaard pc uit te voeren. Deze algoritmes worden daarom uitgevoerd in een aparte processingunit. De processingunit verkleint daarmee de 10Gb-datastroom, zodat de beeldinformatie behapbaar wordt voor de pc. De processingunit hangt met een standaard 1Gb-

ethernet verbinding aan de pc van FEI. Op deze pc draait een softwaremodule die communiceert met de processingunit enerzijds en de pc-software van FEI anderzijds.

### Nieuwe ervaringen

Voor Technolution was het een nieuwe ervaring om fundamenteel en in een vroeg stadium van de productontwikkeling naar fysica en technologieën te kijken; om niet alleen te kijken naar hoe de sensor werkt, maar ook naar de hele TEM daaromheen. Hoe onderdelen en verschijnselen als elektronenstraling, röntgenstraling, vacuüm en trillingen elkaar beïnvloeden. Hoe haal je in die omgeving het beste uit je ontwerp?

Die complexe vragen, op de rand van wat mogelijk is, maken het project extra interessant. Voor de ontwikkeling en productie van de mechanica (de fysieke camerabehuizing) werkt Technolution samen met Demcon. Technolution is de penvoerder in het project en ontwikkelt de elektronica, software en programmeerbare logica. Hierbij wordt gelijk een life cycle management-plan meegenomen, zodat de nieuwe camera de komende jaren goed onderhouden kan worden.

Voor FEI is het totaalplaatje belangrijk, daarom heeft zij intensief contact met haar klanten om continu op de hoogte te zijn van hun wensen en behoeften. Om de klanten nog beter te bedienen, heeft FEI een sterke nieuwe troef in handen met de Falcon elektronendetectorcamera.



/interview klant

# Enexis;

## alles voor een betrouwbaar netwerk

**Netbeheerders zijn verantwoordelijk voor onderhoud en uitbreiding van ons elektriciteitsnetwerk. Deze kostbare infrastructuur wordt maar een klein deel van de dag maximaal benut. Door nieuwe ontwikkelingen zal het netwerk in die piek mogelijk nog hoger belast worden. Om nutteloze investeringen in extra netcapaciteit te vermijden zijn netbeheerders geïnteresseerd in het smart grid voor een betere benutting van het netwerk. Om ervaring op te doen met dit netwerk van de toekomst voert netbeheerder Enexis nu al proeven uit met elektrische auto's en energieopslag.**

Enexis beheert de gas- en elektriciteitsdistributienetten in de provincies Limburg, Noord-Brabant, Groningen, Drenthe, Overijssel en delen van Friesland en Flevoland. Het bedrijf verzorgt voor 2,6 miljoen huishoudens en bedrijven de elektriciteit en voor 1,8 miljoen het gas. Ook beheert Enexis het elektriciteitsnetwerk tussen het hoogspanningsnet en de meterkast thuis. Een netwerk waar de komende jaren veel in zal veranderen. We gaan straks zelf thuis stroom produceren met zonnepanelen en HRe-ketels. Maar we zullen ook veel meer stroom gaan consumeren als we massaal overgaan op de elektrische auto.

“Als je elektriciteit maakt, moet je meteen verbruiken”, vertelt Danny Geldtmeijer, innovator bij Enexis. “En andersom, als je het wilt hebben, dan moet iemand het ergens direct maken.

Je kunt niet alvast een voorraad aanmaken. Er zit flexibiliteit in voor een paar milliseconden. Daarna gaat het net plat, als je niet snel iets regelt.”

### Voorspelbaar

Gelukkig is het menselijk gedrag redelijk voorspelbaar. We starten 's avonds de vaatwasser, zetten koffie en gaan tv kijken. Stroomproducenten en netwerkbeheerders maken van tevoren voorspellingen van het stroomverbruik, legt Geldtmeijer uit. “Op de dag zelf houdt TenneT, beheerder van het hoogspanningsnet, bij wat er gebeurt. Zij hebben reservevermogen achter de hand om bij of af te regelen. Daarnaast draaien centrales nooit helemaal op vol vermogen. Elke centrale kan zijn productie snel verhogen of eventueel verlagen, om de actuele vraag exact te kunnen volgen.”

### Netwerk zwaarder of slimmer

In de toekomst, met elektrische auto's, windmolens en zonnepanelen, zal dat een stuk minder voorspelbaar worden. Zowel de vraag als het aanbod zal sterk fluctueren. Enexis kijkt dan ook met grote belangstelling naar de elektrische auto, zegt Geldtmeijer. “Als elektrisch vervoer serieus doorbreekt, met tientallen procenten, moeten we het netwerk mogelijk flink verzwaren. Dat betekent stoepen openbreken om duizenden kilometers nieuwe kabels te leggen. Of we moeten ons huidige net slimmer gebruiken. Want nu wordt alleen in de avond het net maximaal belast, maar de rest van de tijd hebben we ruimschoots capaciteit over. We hebben nog zoveel ruimte om energie te transporteren in de daluren. Dat willen we dus wat slimmer doen: spreiden. Dan kom je vanzelf bij smart grids.”

Zo'n slim elektriciteitsnet is ook nodig voor duurzame energie. Immers, windmolens maken elektriciteit als het waait, niet op aanvraag als je het nodig hebt. We gebruiken veel energie 's avonds, maar zonnepanelen produceren alleen overdag. Enexis ziet de elektrische auto als een oplossing voor dit probleem. “De accu van de elektrische auto kun je gebruiken om de restcapaciteit (van windmolens en zonnecellen) naar door te sluizen. En in pieksituaties, als er eens meer dan 100% vraag is, kun je in de toekomst zelfs elektriciteit terug halen uit die accu.”

### Elektrische auto

Ideeën genoeg, maar hoe pakt dat uit in de praktijk? Om daar achter te komen, heeft Enexis vijftien Volkswagen Golf stationwagens laten ombouwen tot volledig elektrische wagens. Zelf bracht het bedrijf kennis in voor het opladen van de accu's. Er is bewust gekozen voor een ruime auto. "De eerste puur elektrische auto's zijn kleine tweepersoons- of sportauto's. Daar kun je je als modale autorijder niet mee identificeren. Daarom hebben we voor een gezinsauto gekozen. We zijn geïnteresseerd in hoe zo'n elektrische auto bevalt. Gebruik je hem binnen je bedrijf? Maar ook wat de kans van slagen is dat consumenten dit gaan doen. Wat willen mensen met zo'n auto? We monitoren hoe ze rijden en hoeveel stroom ze gebruiken. Want daaruit kun je weer destilleren wat we voor ons netwerk kunnen verwachten."

### Flexibiliteit

Enexis ontwikkelt alleen op conceptueel niveau. Daarna zoekt de beheerder een bedrijf met expertise om de benodigde technologische oplossingen te maken. "We zochten een combinatie van hardware en software. Daar bleek Technolution heel goed in te zijn. En we hebben flexibiliteit gevonden bij Technolution. We kunnen op basis van een concept samen aan de slag. Dat begrijpen we samen en gaandeweg het traject wordt steeds helderder wat het resultaat wordt. Technolution biedt ondernemerschap en inhoudelijke kennis, waardoor het steeds weer op zijn pootjes terecht komt."

In een eerste verkenning leverde Technolution de MobiBoxx en sensoren om in de elektrische auto's het verbruik en de efficiëntie te meten. Het tweede project draaide om het snel en slim opladen van de auto's. "Je plukt de auto in op de laadpaal. Doormiddel van een app op je mobiele telefoon geef je aan hoe laat je weer wilt gaan rijden en hoeveel kilometer je dan wilt rijden. Het achterliggende systeem kijkt voor alle aangesloten auto's wat de juiste tijd is om te gaan laden en met welke stroomsterkte."

### Het smart grid voor Enexis

Geldtmeijer ziet nog geen eenduidige definitie voor 'het smart grid'. "We hebben het nu vooral over welke functionaliteit eronder valt. De essentie van smart grid is informatie beschikbaar maken en daar vervolgens slim mee handelen; slimme apparatuur, zoals een wasmachine die pas start als er een overschot aan elektriciteit is, bijvoorbeeld van de zonnepanelen. Je voegt een informatielaag toe aan je elektriciteitsnet, je gaat meten wat er aan de hand is en daar probeer je vervolgens op te sturen. Bijvoorbeeld met een signaal 'probeer nu te minderen, want straks is het bijna voor niks'. Maar het duurt nog wel een paar jaar voor het op grote schaal zover is."

De kernvraag draait om geld. "Als wij kijken wat er allemaal op ons af komt, dan kunnen we voor tientallen miljarden nieuwe kabels gaan aanleggen of voor enkele miljarden een smart grid realiseren om die investeringen te vermijden. Willen consumenten zich flexibeler opstellen, ondersteund met een smart grid? Kunnen we technologie ontwikkelen, waarmee consumenten ook dan van alle gemakken voorzien blijven? Of willen ze flink meer gaan betalen? Zodra we antwoord hebben op deze vragen kunnen we goed inspelen op de vraag van de markt. Onze gezamenlijke projecten met Technolution helpen bij het beantwoorden van die vraag."

Bekijk de video 'Praten met de auto' waarin Danny Geldtmeijer de proef met elektrisch rijden zelf uitlegt.

<http://www.youtube.com/EnexisSlim>



Danny Geldtmeijer,  
innovator bij Enexis



/trends &amp; hypes

# Softwarekeurmerken: een trend of hype?

&gt;&gt;naast verificatie ook validatie noodzakelijk



**Wie software laat ontwikkelen, wil een werkend systeem geleverd krijgen. Maar hoe kun je na oplevering weten of het systeem goed is ontwikkeld? Een keurmerk kan hier een basis van zekerheid voor bieden. Hoe gaat softwareontwikkeling in zijn werk en wat toetst een softwarekeurmerk precies?**

De ontwikkeling en productie van maatwerksoftware wordt steeds vaker uitbesteed. Dan gaat het niet over een simpele teksteditor, maar over complexe systemen voor banken, ziekenhuizen, verkeersmanagement en dergelijke. Na een functionele aanbesteding wil de opdrachtgever bij oplevering zeker weten dat de software doet wat het moet doen. Vervolgens wordt het systeem jaren gebruikt en onderhouden. Op termijn kunnen nieuwe wensen ontstaan, waardoor de klant extra functies aan het systeem wil toevoegen. Als de software goed is gemaakt, is het relatief eenvoudig om nieuwe functionaliteit toe te voegen. Daarvoor moet de code duidelijk, helder en overzichtelijk zijn. Niet alleen voor degene die de code heeft geschreven, maar ook voor andere ontwikkelaars. Kan je een muurtje bijbouwen zonder dat het hele gebouw in elkaar stort?

## Hulpmiddelen

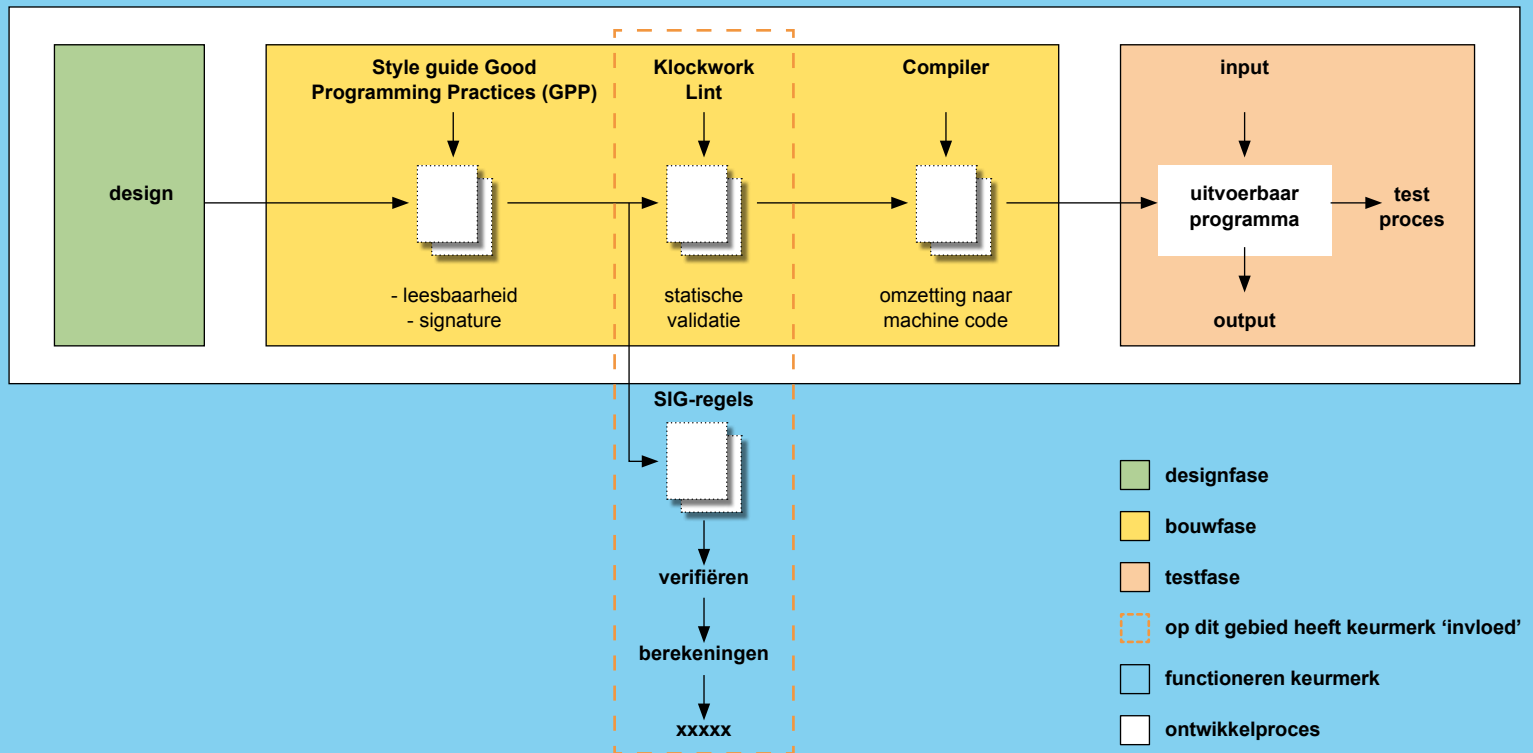
De structuur van de software en de manier waarop het geschreven is (de stijl) bepalen in grote mate de onderhoudbaarheid. Afzonderlijke functies dienen herkenbaar te zijn en snel aanpasbaar, zonder andere functionaliteit te beïnvloeden. Gelukkig bestaan er de nodige hulpmiddelen om duidelijke code te schrijven. Zoals er grammatica en stijlgidsen zijn voor taal, bestaan er regels en afspraken voor het correct opstellen van code. Die op hun beurt weer verwerkt zijn in speciale controleprogramma's als Klockwork en Lint. Deze gereedschappen analyseren de code op bekende valkuilen. "Ik snap wat de code moet doen, ik zie dat je hier een kast maakt van 10 planken en ik zie dat je iets op plank 11 probeert neer te leggen. Wees je daarvan bewust, doe daar wat aan."

## Stijl en complexiteit

Andere aspecten waar de hulpprogramma's naar kijken, zijn de stijl en de grootte van de modules. Voor een computer maakt het niet uit hoe de code eruit ziet, zolang alles maar klopt. Maar goede ontwikkelaars zorgen voor overzichtelijkheid. Daar bestaan regels voor (samengevat in de Good Programming Practices) en stijlrichtlijnen. Bijvoorbeeld dat code die functioneel bij elkaar hoort, ook bij elkaar in één module staan. Zo worden functies gedefinieerd die vanuit het hoofdprogramma zijn aan te roepen. Wordt een functie te groot, dan knip je die weer op in aparte functies. Hoe complexer de code, hoe groter de kans op fouten. Een analyse met die insteek telt het aantal beslissingen per functie; de 'if\_then' vragen. De analyse waarschuwt als dat er teveel zijn. Het compileren van de code is ook een controle. Pas wanneer alles volgens de regels van de desbetreffende taal is opgeschreven, zet de compiler de code om in machinetaal. De software is nu klaar om te gebruiken. Dan komen de functionele tests: doet het programma wat het moet doen?

## Onafhankelijk keurmerk

De klant weet nu dat alles werkt. Maar welke kwaliteit heeft de software? Is de code later makkelijk te onderhouden en aan te passen? Dat is lastiger te bepalen. Precies in die niche is de Software Improvement Group (SIG) gedoken. Dit Nederlandse keuringinstituut geeft een onafhankelijk oordeel over de kwaliteit van softwareproducten in de vorm van een keurmerk volgens de ISO/IEC 9126 normering. Gekeken wordt naar de kwaliteit van



het ontwerp en het gevolgde proces tijdens de ontwikkeling. De architectuur wordt geanalyseerd, waarbij bijvoorbeeld gekeken wordt of de functionaliteit is opgedeeld in modules die begrijpbaar zijn. Maar ook naar hoe de ontwikkelaar tot het ontwerp is gekomen. Daartoe worden interviews met systeemexperts, de architecten en ontwerpers van de code gehouden: wat was je gedachtegang? Hoe heb je het opgesplitst? Wat moet het systeem doen en hoe heb je dat bereikt? En op welke manier test de designer zijn eigen ontwerp? Deze informatie wordt meegenomen bij het bekijken van de code, zodat geëvalueerd kan worden of deze gedachtegang en overwegingen terug te vinden zijn in het eindresultaat.

De kracht van het softwarekeurmerk is de vertaling van complexe code naar een eenduidig kwalificatiemiddel. Een groot systeem kan uit wel duizend modules bestaan die samen een softwareproduct vormen. Die worden door tientallen soorten tests gehaald. Zo veel resultaten kan niemand meer overzien. Het keuringsinstituut vertaalt deze hoeveelheid aan informatie tot een score van één tot vijf sterren, met daarbij eventueel een advies voor verbeteringen.

### Goede code = flexibiliteit

Onderhoud is vaak de grootste kostenpost van een systeem, maar wordt nog wel eens over het hoofd gezien bij een uitbesteding. Hoeveel moeite het kost om later nog iets aan te passen, hangt af van de onderhoudbaarheid. In complexe, slecht onderhoudbare code is het veel moeilijker om een aanpassing in te voegen.

Na elke aanpassing is veel testwerk nodig om zeker te weten dat er geen andere functionaliteit is beïnvloed. Met goed onderhoudbare code kan een ontwikkelaar meteen naar de juiste module gaan, waar hij in een compact stukje code een wijziging aanbrengt.

### Vakmanschap

Softwarekeurmerken zijn een trend, ze bieden een controlehandvat voor het uitbesteden van complexe softwaresystemen. De hoofdtaak van de klant ligt bij het verzorgen van een oplossing en aan welke functionele eisen deze oplossing moet voldoen. Hoe dit technologisch is opgebouwd, wordt niet tot in detail vastgelegd. Wat vast staat is dat het systeem aan een aantal eisen moet voldoen met betrekking tot onderhoud- en uitbreidbaarheid. Dan biedt een softwarekeurmerk een uitkomst om dit niveau te toetsen. Klanten van Technolution experimenteren met het softwarekeurmerk en hebben meerdere systemen die in ontwikkeling waren via de SIG beproefd. Technolution was niet bekend met de kwalificatiewijze van SIG, maar scoorde desondanks hoog in de SIG-ranking.

Uiteindelijk verifieert het softwarekeurmerk dus of de code netjes en overzichtelijk is geschreven, maar wordt er niet gevalideerd of het opgeleverde systeem ook doet wat ervan verwacht wordt. Daarvoor heb je nog steeds vakmensen nodig die met een gedegen ontwikkelstrategie en meerdere tussentijdse testen de voortgang en het uiteindelijke doel bewaken.



## /medewerker aan het woord

# Enno Romkema

### \*kennis is essentieel voor ons bedrijf

**Enno Romkema is Chief Technology Officer (CTO) van Technolution en een van de oprichters. Zijn passie voor technologie is nog steeds zijn grootste drijfveer.**

“Ik ben in 1979 afgestudeerd aan de hts in Leeuwarden. Na een korte periode bij Olivetti ben ik bij Brinkman komen werken. Dat bedrijf maakte besturingssystemen voor de tuinbouw. Heel interessant, maar het was een product-organisatie. Ik had veel bredere aspiraties. Ik wilde meer variatie in onderwerpen en voor verschillende markten werken. Bij Brinkman trof ik mensen met dezelfde wens: Jan, onze huidige algemeen directeur, en Rob, inmiddels met pensioen. Toen hebben we in 1987 Technolution opgericht. Een projectorganisatie, waarin we steeds dat konden doen wat we het leukst vonden: projecten met snelheid en afwisseling.

#### **Passie voor techniek**

Technologie is cruciaal voor ons bedrijf. We hebben niet voor niets de naam Technolution gekozen: technology solution. De ontwikkeling van het bedrijf is dan ook nauw verbonden met de ontwikkeling van technologie. De tijdgeest van 1987 was een totaal andere dan de huidige. Het was een stuk overzichtelijker; de pc had net zijn intrede gedaan. Apparatuur stond nog op zichzelf, het communiceerde niet of nauwelijks met elkaar. De professionele elektronica en software was een jonge, beginnende bedrijfstak waar iedereen zelf dingen bedacht. Er waren weinig standaarden, de IBM-pc uit 1981 was een eerste grote standaard. De microprocessor bestond al wel en die was essentieel voor wat wij wilden gaan doen: industriële systemen bouwen. Inmiddels kan alles draadloos. De evolutie in componenten heeft ons zelfs een derde tak van technologie bezorgd: de programmeerbare logica.

#### **Technologiemanagement**

Vroeger kon ik als technisch directeur alles overzien. Met het voortschrijden van de techniek is het onmogelijk alles te weten, dus word ik ondersteund door collega's. We hebben nu vijf technologiemanagers die elk een specialisme beheersen. Zij houden de ontwikkelingen op hun vakgebied in de gaten. We willen vooruitstrevend zijn in technologie. Dat verwachten klanten ook van ons. Nog voordat een klant ergens om vraagt, willen wij onze kennis op orde hebben. Dat doen we met interne projecten om kennis van en praktische ervaring met nieuwe technologie op te doen. Dat hebben we gedaan met smartcards, met de 10Gb link die nu onder andere in de Falcon elektronenmicroscoop voor FEI zit en zo is ook onze MobiBoxx voor in-carautomatisering ontstaan. Het zijn eigen investeringen die we kunnen gebruiken in projecten. We willen niet zelf de markt op gaan met een product, anders zouden we een concurrent van onze eigen klanten kunnen worden. We willen projecten uitvoeren en maatwerk leveren.

#### **Kennisopbouw**

Kennis is essentieel voor ons bedrijf. We voeren een bewust actief beleid om kennis te delen. Want kennis verspreidt zich niet automatisch naar het punt waar het het hardst nodig is. Je moet het actief verspreiden en je moet mensen de tijd en gelegenheid geven. Dat doen we onder andere met Technology EXchange meetings: dat zijn door de technologiemanager georganiseerde bijeenkomsten waarin iemand een technologie uitlegt aan collega's. Kennis is het enige dat meer wordt als je het deelt. Het vraagt een bepaalde mentaliteit van onze medewerkers: openheid en gretigheid om te willen leren. En omgekeerd wil ik dat als technisch directeur ook uitstralen. Ik vind het een eer en genoeg om mede leiding te geven aan deze technologie gedreven organisatie.”

## /colofon

Objective is een uitgave van Technolution B.V. Alle eerder verschenen uitgaven van Objective zijn te downloaden op [www.technolution.eu/objective](http://www.technolution.eu/objective)

#### **Verspreiding**

Controlled circulation voor relaties van Technolution

#### **Technolution B.V.**

Zuidelijk Halfroond 1  
P.O. Box 2013  
2800 BD GOUDA  
The Netherlands

T +31(0)182 59 40 00  
E [info@technolution.eu](mailto:info@technolution.eu)  
I [www.technolution.eu](http://www.technolution.eu)