

Automatische software-analyse in context

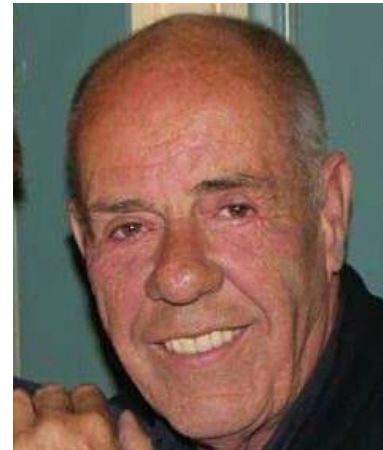


@jurgenvinju
#swanalyse

Jurgen J. Vinju

Intreerede
12 februari 2016

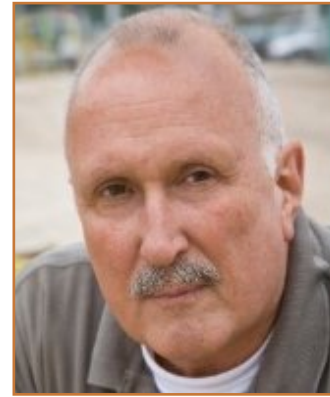
Familie Bedankt!



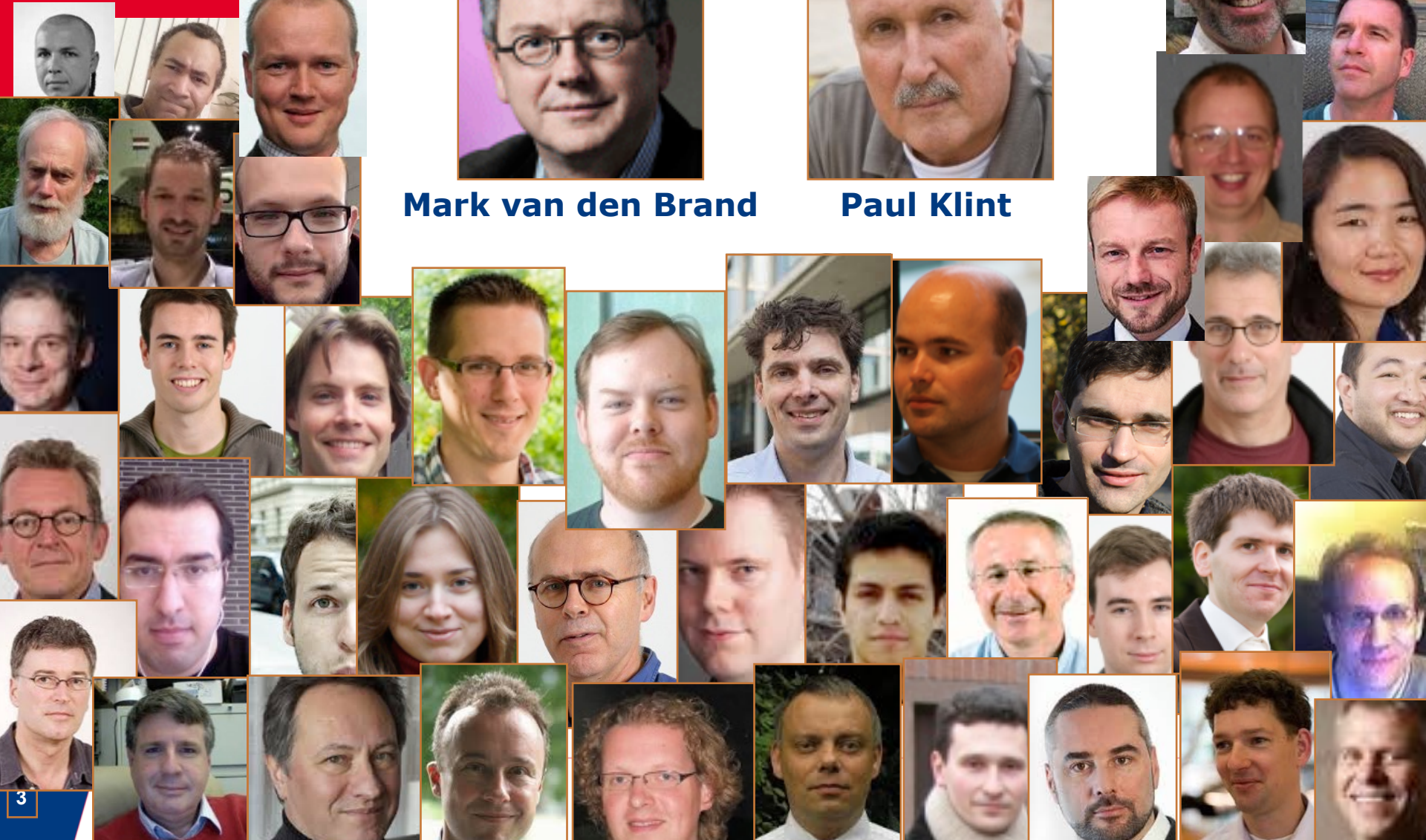
Collega's Bedankt!



Mark van den Brand



Paul Klint



**Werkgevers
&
subsidie-
vertrekkers
bedankt!**



Centrum Wiskunde & Informatica



**College “Inleiding
Automatische
Software Analyse”**

familie &
vrienden

**Agenda “kansen en
uitdagingen in
software analyse”**

collega’s software
engineering

of je hebt zo’n dag...

**Gewoon even
wachten op de
borrel**

collega’s uit
industrie, onderwijs & onderzoek

Uitnodiging tot samenwerken

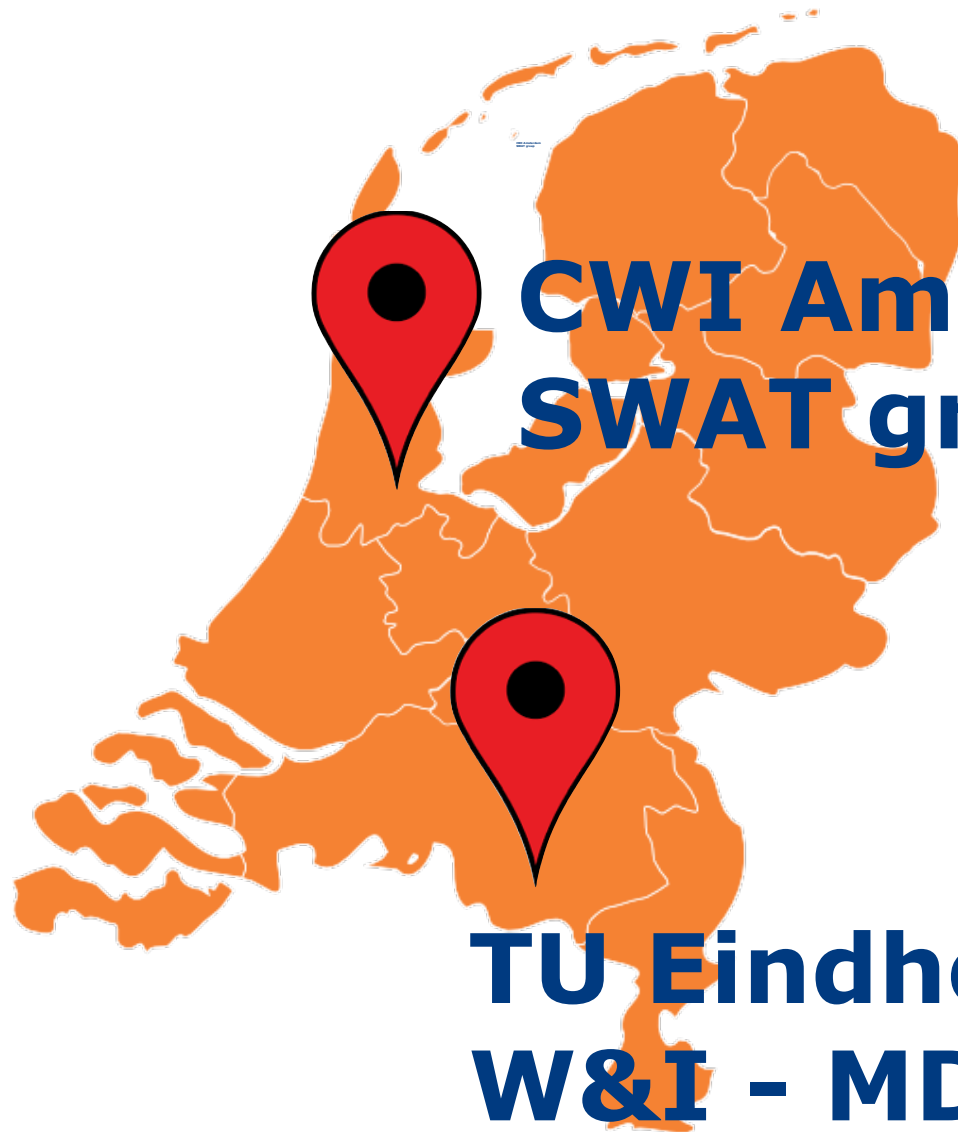
geïnteresseerde master studenten
promovendi, post-docs, tenure trackers

Promotie veld software analyse

politici,
beleidsmakers,
entrepreneurs

**rationeler
omgaan met
software**

Menu



**CWI Amsterdam
SWAT groep**

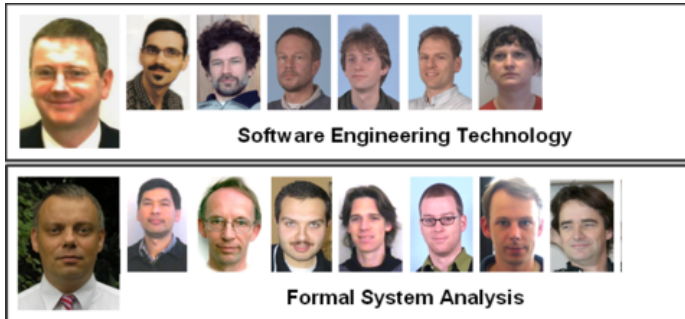
**TU Eindhoven
W&I - MDSE sectie**

**TU Eindhoven
W&I - MDSE sectie**

**CWI Amsterdam
SWAT groep**



"Automated Software Analysis"

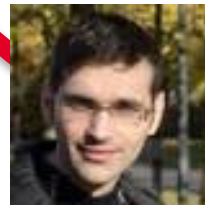


TUE - MDSE



woensdag

CWI - SWAT



vrijdag



NWO Big Software “MERITS”

*Jan Friso Groote, Joost Gabriels
Arjan Mooij (TNO-ESI), Gernot
Eggen (Philips Healthcare)*

Vacature PhD

**CWI vacature senior
post-doc faster
modular static analysis**

EU H2020 en ECSEL

*voorstellen: Arjan Mooij
(TNO-ESI), Gernot Eggen
(Philips Healthcare)*

Samenwerking

**“Empirical analysis of the relationship between
CC and SLOC” (JSEP 2016, ICME 2014)**
Davy Landman, Alexander Serebrenik, Eric Bouwers

TUE Impulse 2 project met OCÉ

*Sander Stuijk, Twan Basten, Lou
Somers, Henk Corporaal,
Mark van den Brand*

Vacature PhD

CWI PPS met ING bank

*Jouke Stoel, Rob van der Mei, Joost
Bosman jr, Joost Bosman sr. (ING)*

Post-doc op CWI (FM & SWAT)

Yanja Daysuren, Frank de Boer



1. Wat is software?

DE
MAATSCHAPPIJ
DAT BEN **JIJ.**

**2. Software leren
analyseren is belangrijk!**



Waar gaat het eigenlijk over?



**3. Goed analyse
gereedschap!**



**4. Veel meer
"contextueel"
slim gereedschap nodig...**

Software in de media en de vergaderzaal



Magisch, raar, eng.



Uitvoerend, gewoontjes

**Software is niet magisch of triviaal.
Software is “gewoon” het product van creativiteit.**

[illegible]

roman

software

Publieksvragen (handen zien)

- Wie schrijft of leest broncode?
 - scripts, programma's, systemen?
- Wie schrijft natuurlijke taal?
 - boeken, rapporten, papers?
- Wie schrijft (blad)muziek?
 - pop, jazz, klassiek?

Een heel klein stukje software

“Postscript” (zit in PDF)

```
%! PS EPS
3{25 34 moveto
25 -34 lineto
17 -38.2 lineto
17 20 lineto
-17.6 0 lineto
120 rotate
}repeat stroke showpage
```

taal om heel
netjes mee te
schrijven,
tekenen en
rekenen

door Kees
van der Laan
(1943-2015)

Wie weet wat dit programma tekent?

```
%! PS EPS
```

```
3{25 34 moveto
```

```
25 -34 lineto
```

```
17 -38.2 lineto
```

```
17 20 lineto
```

```
-17.6 0 lineto
```

```
120 rotate
```

```
}repeat stroke showpage
```

uit het
hoofdje



Broncode begrijpen

- PS/PDF document = broncode
- Broncode “definieert de toekomst”
 - Wat de computer precies gaat doen
 - Geduldig en heel snel en elke keer weer
- Alles kan, alles mag!

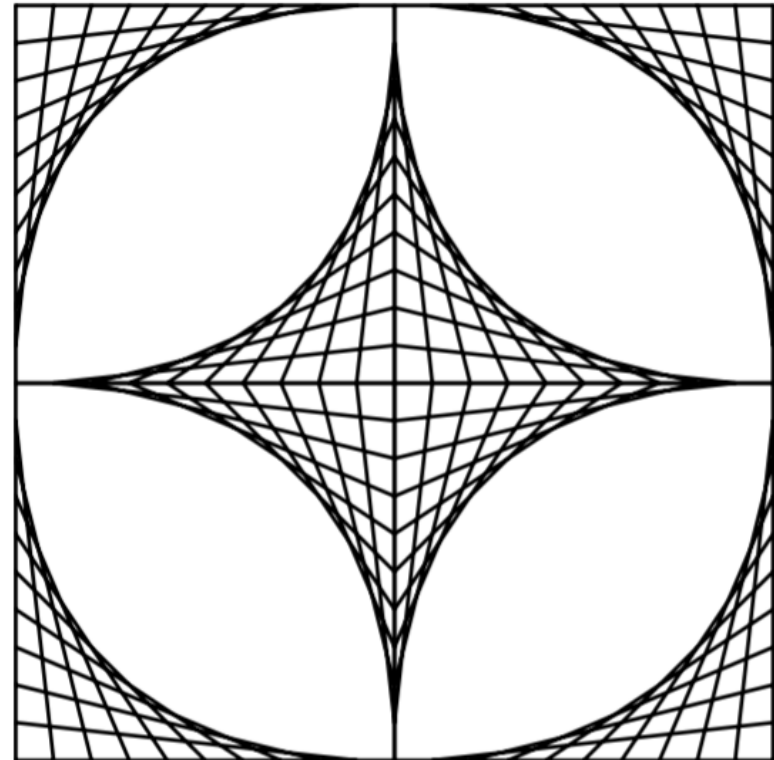
```
%! PS EPS
3{25 34 moveto
25 -34 lineto
17 -38.2 lineto
17 20 lineto
-17.6 0 lineto
120 rotate
}repeat stroke showpage
```

Alles kan

```

/envelopdict 9 dict def
/envelope{envelopdict begin %tilxviii
0 setlinejoin 1 setlinecap
/r 100 def /n 5 def /mn n neg def
/hr .5 r mul def /mhr hr neg def
/h hr n div def
/tile{2{mn 1 n{/y exch h mul def
      hr y moveto y neg hr lineto}for
      180 rotate}repeat
}def
mhr r hr{/i exch def
mhr r hr{/j exch def
  gsave i j translate
    i j mul 0 lt{90 rotate}if
    tile stroke
  grestore
}for}for
end}def %envelope

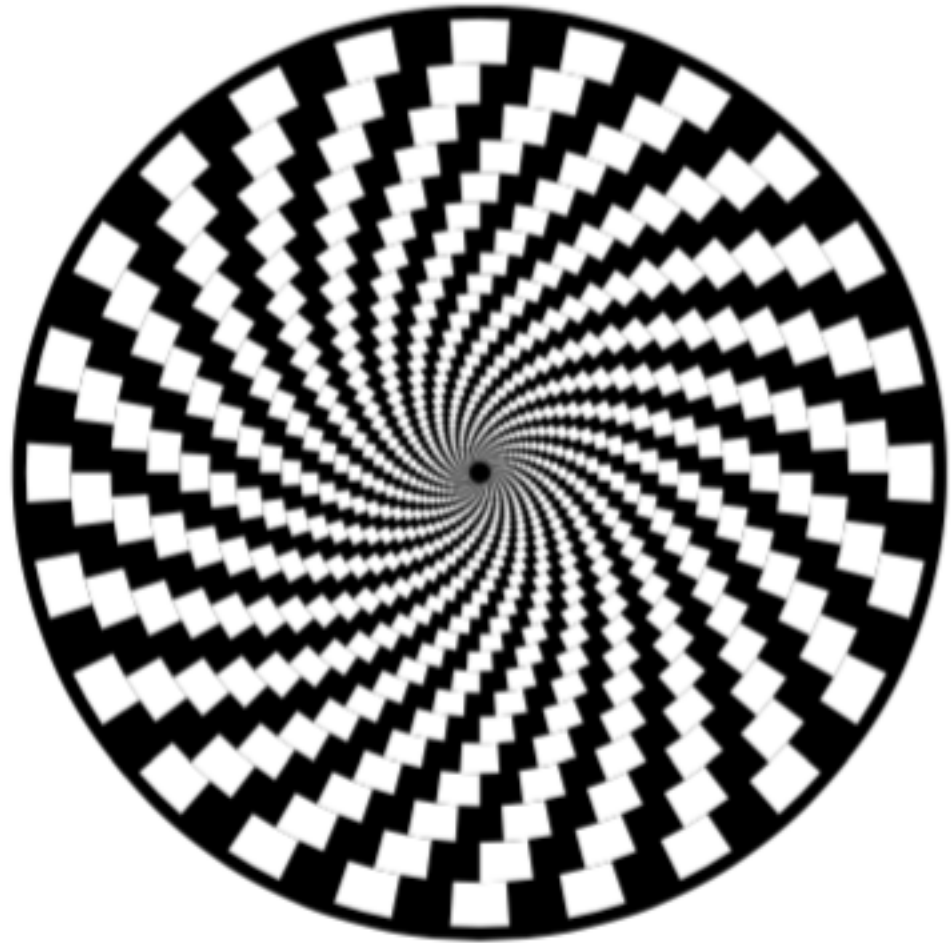
```



[Kees van der Laan]

Alles mag

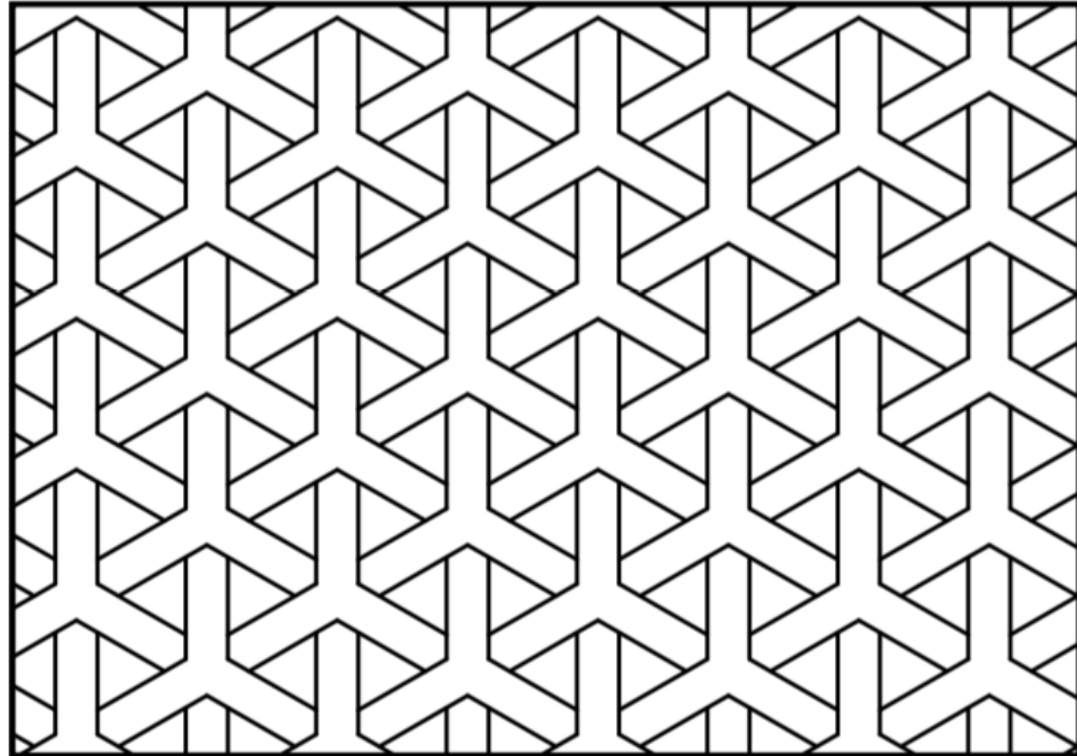
```
/square{rin 3.75 cos mul rin -3.75 sin mul moveto
  0 0 rout -3.75 3.75 arc
  0 0 rin 3.75 -3.75 arcn fill
}def
/ring{/rin rout 1 1.5 3.75 sin mul sub mul def
  24{gsave square grestore 15 rotate
    }repeat /rout rin def
}def
```



[Kees van der Laan]

Mooi oppervlakkig

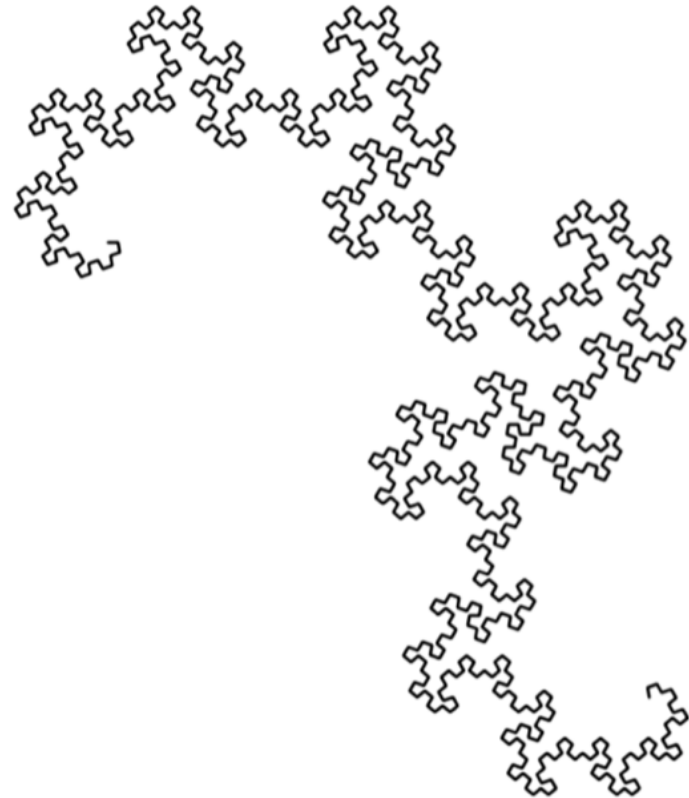
```
/fencedict 16 dict def
/fence{fencedict begin %tilxvii
0 setlinejoin 1 setlinecap
/s 30 def /ms s neg def /ss s 1.732 mul def
/hs s 2 div def /mhs hs neg def
  /hss hs 1.732 mul def
/w s 3 div def /wds w 1.732 div def
/hw w 2 div def /hwds hw 1.732 div def
/line{hw s hwds sub moveto
  hw hwds lineto
}def
/tile{3{gsave line -1 1 scale line stroke
  grestore 120 rotate}repeat
}def
/pattern{gsave
6{gsave
  7{gsave tile
    hw hss add hs hwds add translate
    tile
    grestore
    w ss add 0 translate
  }repeat
  grestore 0 s wds add translate
}repeat grestore
```



[Kees van der Laan]

Lekker diepzinnig

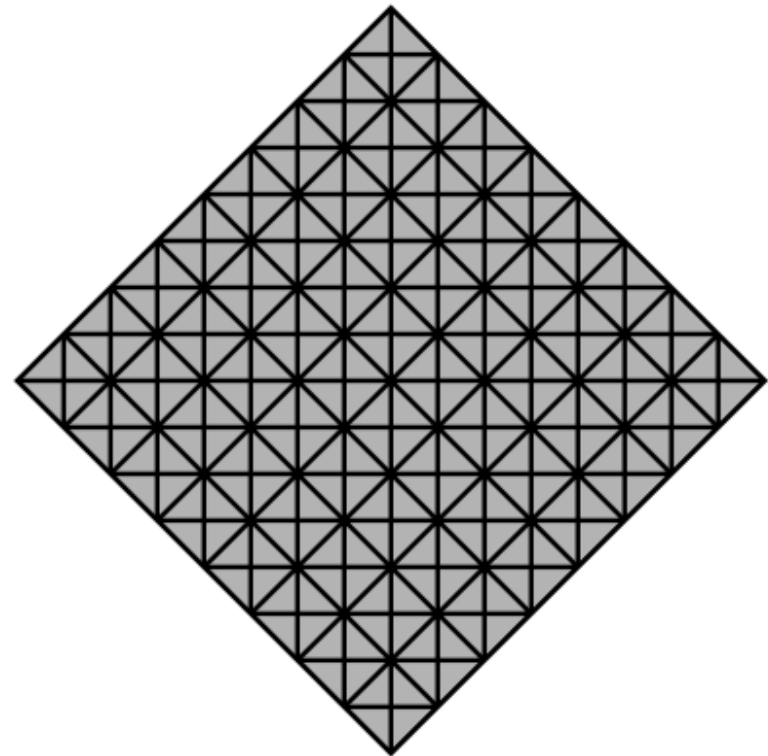
```
/draakdict 10 dict def
/draak %dragon curve with angle A and order p, global scaling
    %angle p ==> dragon curve
{draakdict begin /p exch def /b 180 3 -1 roll sub def %angle B
/s {100 mul} def /h 2 p -2 div exp s def %size piece scaled
/s 0 def
0 0 moveto h 0 lineto
1 1 2 p exp 1 sub{%for n
    /m exch def
    {m cvi 2 mod 0 eq {/m m 2 div def}{exit}ifelse}loop
    m cvi 4 mod 1 eq {/d 1 def}{/d -1 def}ifelse
    /s s d sub def
    h s b mul cos mul h s b mul sin mul rlineto
}for %n
end}def
100 10 draak stroke showpage
```



[Kees van der Laan]

Helemaal abstract

```
/e 10 def /r e 8 mul def /mr r neg def
/frame{r 0 moveto 0 r lineto mr 0 lineto
  0 mr lineto closepath
}def
/lines{-7 1 7{e mul dup mr moveto
  r lineto
}for stroke
}def
/diagonals{2 2 15{e mul dup r moveto
  neg mr exch lineto
}for stroke
}def
gsave
frame gsave .95 setgray fill grestore clip
lines diagonals 90 rotate lines diagonals
grestore frame stroke %showpage
```



Kees van der Laan

Lozenge - Piet Mondriaan

Super gedetailleerd



L. Peter Deutsch - author of "Ghostscript" (postscript interpreter)



C:\> Software heeft geen
beperkingen t.o.v. onze
creativiteit.

C:\> We schrijven zinnige
software en we schrijven
absurde software.

C:\> We schrijven zelfs
zinnige software die niet
te vatten is.

Software uitersten



volume

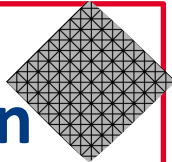
**Gedetailleerd en
oppervlakkig
= Data**



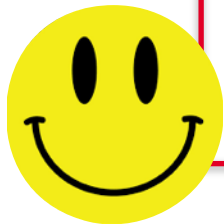
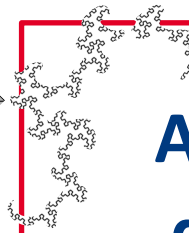
**Gedetailleerd en
diep- of onzinnig
= Complex**



**Abstract en
oppervlakkig
= Overzichtelijk**



**Abstract en
diepzinnig
= Inzichtelijk**

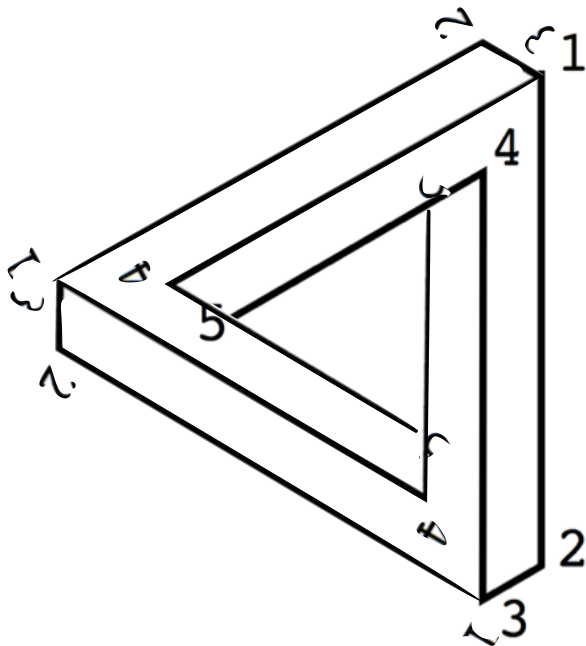


betekenis

Maar is het effect van deze code?

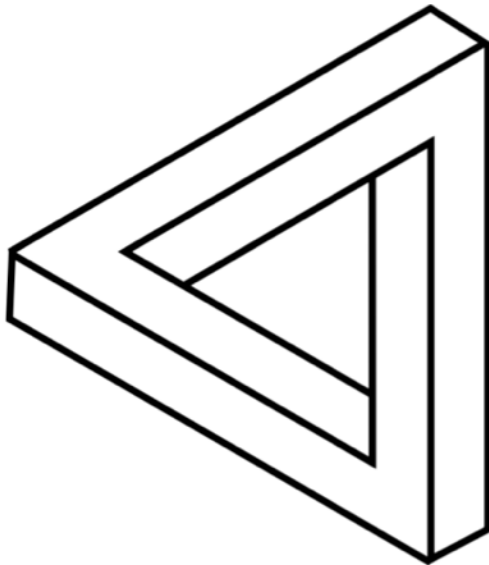
```
%! PS EPS
3{25 34 moveto
25 -34 lineto
17 -38.2 lineto
17 20 lineto
-17.6 0 lineto
120 rotate
}repeat stroke showpage
```

Analyse



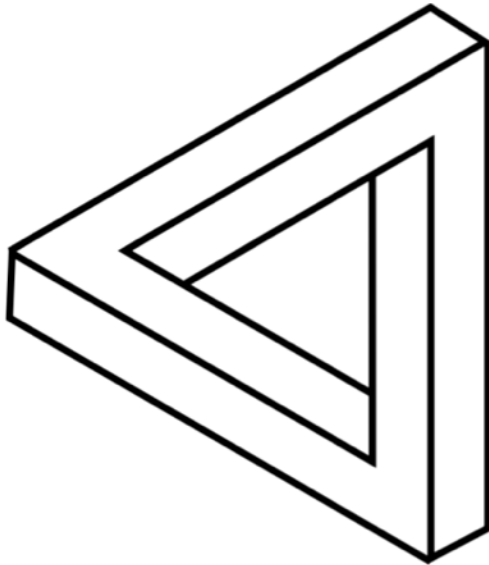
```
%! PS EPS
3{25 34 moveto
25 -34 lineto
17 -38.2 lineto
17 20 lineto
-17.6 0 lineto
120 rotate
}repeat stroke showpage
```

Analyse



```
%! PS EPS
3{25 34 moveto
25 -34 lineto
17 -38.2 lineto
17 20 lineto
-17.6 0 lineto
120 rotate
}repeat stroke showpage
```


Vragen stellen over software



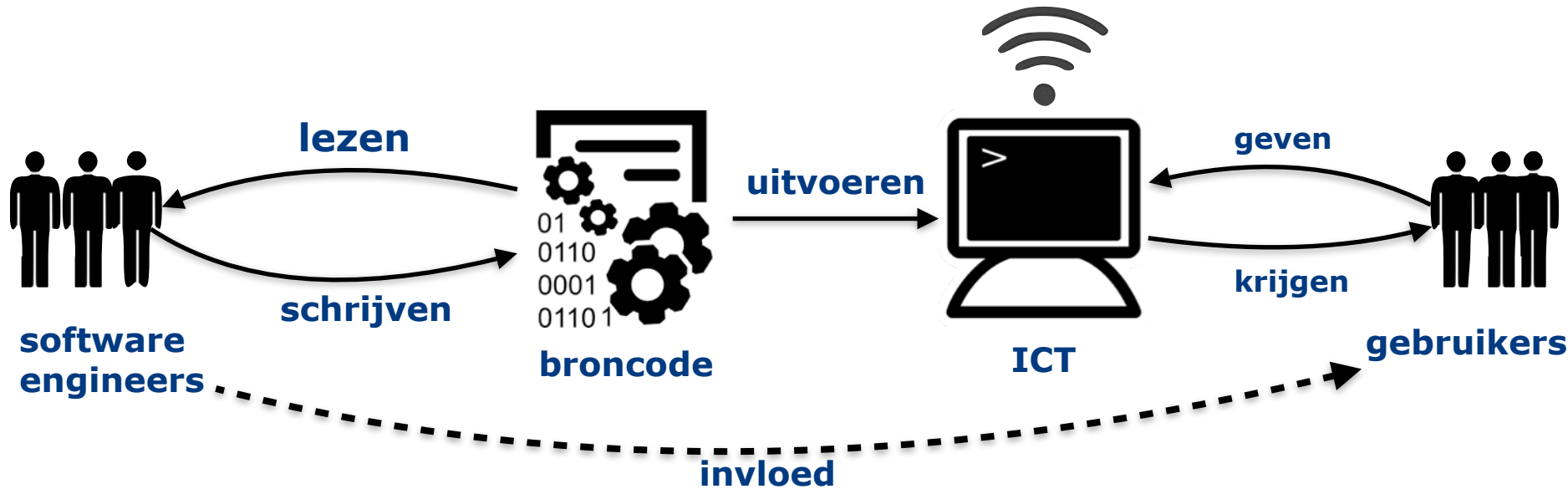
[Penrose, Escher]

- Simpele code
- Simpele vraag:
 - welk plaatje?
- Complexe analyse:
 - draaisymmetrie
 - simuleer het maar...
- Antwoord: **absurde kunst**

Broncode begrijpen

- Creativiteit + Schrift = Literatuur
- Creativiteit + Bladmuziek = Concert
- Creativiteit + Broncode = Software

- Absurd of zinnig?
- Nuttig of geinig?
- Voorspelbaar of oncontroleerbaar?
- Dat hangt er van af...
- Broncode analyse = begripend lezen



Software is niet het medium.
Software is de boodschap.

Wie broncode schrijft

- definieert de toekomst
- maakt én beperkt mogelijkheden van mensen
- laat een wet uitvoeren (of ontwijkt de wet juist)
- legt de regels van het spel vast
- uit een (politieke) mening
- bespaart óf verspilt energie
- heeft (onbewust) macht over mensen en instanties
- heeft verantwoordelijkheid voor personalia en privacy
- De boodschap is heel **gevarieerd**
- De boodschap is nogal **impliciet**

Concreet voorbeeld: formulieren

Bent u man of vrouw?

- ☒ Man
☐ Vrouw

Bent u man of vrouw?

- ☒ Man
☒ Vrouw
☐ Waarom wilt u dat weten?
☐ Weet ik niet
☐ Other:

Bent u man of vrouw?

M/V?

Bent u man of vrouw?

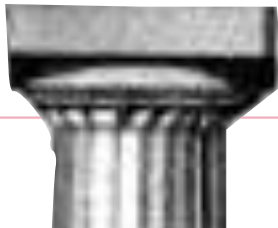
1 2 3 4 5 6 7 8 9 10

Man ☐ ☐ ☒ ☐ ☐ ☐ ☐ ☐ ☐ ☐ Vrouw

Broncode beïnvloedt ons *vaak ongemerkt* heel direct

*Broncode analyse is noodzakelijk
omdat we antwoorden willen
op gewone vragen over de software
die we werkelijk overal voor gebruiken.*

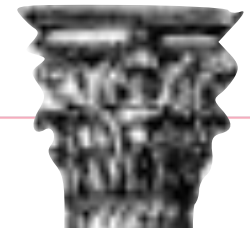
functies?



kwaliteit?



kosten?



Simpel gevraagd, moeilijk beantwoord

- Wat doet deze software?
 - Voert het de wet, het beleid, de regels uit?
 - Vertelt het de waarheid?
 - Lekt het informatie?
 - Verbruikt het veel energie?
- Hoe werkt deze software?
 - Waarom is het correct?
 - Wat is er fout?
 - Kan het aangepast worden (sneller, kleiner)?
- Wat kost deze software?
 - Bij normaal gebruik?
 - Bij nieuwbouw?
 - Bij uitbreiding?
 - Bij renovatie?



Software analyse: doen!

- Software kunnen vertrouwen:
 - niet minder doen dan verwacht
 - niet meer doen dan verwacht
- Software-kosten kunnen beramen:
 - verborgen gebreken vermijden
 - gebruiks- en onderhoudskosten
- Software weten aan te passen
 - niet meer *per ongeluk werkend*
 - maar *expres goed* gedaan





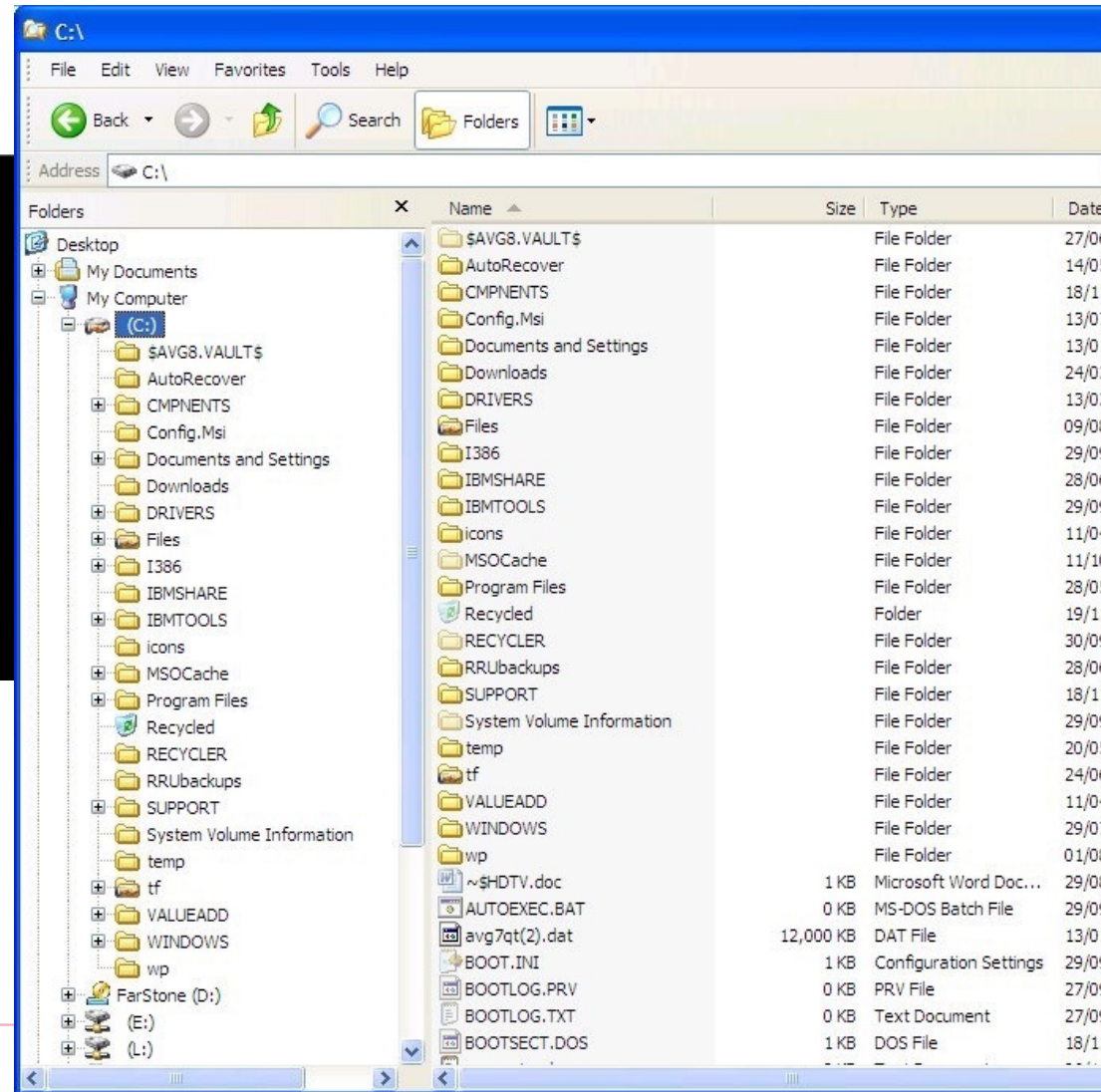
Software analyseren is echt moeilijk.

Want echte broncode is *bovenmenselijk* complex

C:\> dir

```
C:\>dir
Directory of C:\.

.                <DIR>          00-00-2000  14:33
..               <DIR>          01-01-1900   0:00
BT1              <DIR>          13-01-2000  19:27
BT2              <DIR>          14-05-2000  21:17
CIU              <DIR>          18-10-2000  16:43
DARKSUM          <DIR>          12-07-2000   9:01
DUNE2            <DIR>          12-07-2007  15:52
KINGSQ~1         <DIR>          23-04-2000  17:28
STAR             <DIR>          19-01-2007  23:06
UFO-EN~1         <DIR>          19-01-2007  23:06
UFO-EN~2         <DIR>          19-01-2007  23:06
UNDERW~1         <DIR>          23-04-2000  17:30
XCOM             <DIR>          19-01-2007  23:04
DOSSTUFF.TXT     375 23-04-2000  18:48
               1 File(s)      375 Bytes.
               13 Dir(s)     110,540,800 Bytes free.
```



Een lijst met bestanden

ls.c

3894 regels
77 pagina's
541 keuzes



echte software is groot, heterogeen en ingewikkeld

Echte software is groot

Bestandenlijst

5000 regels code

Stemmen

70.000 regels code

MRI scanner

1 miljoen regels code

Bank

20 miljoen regels code

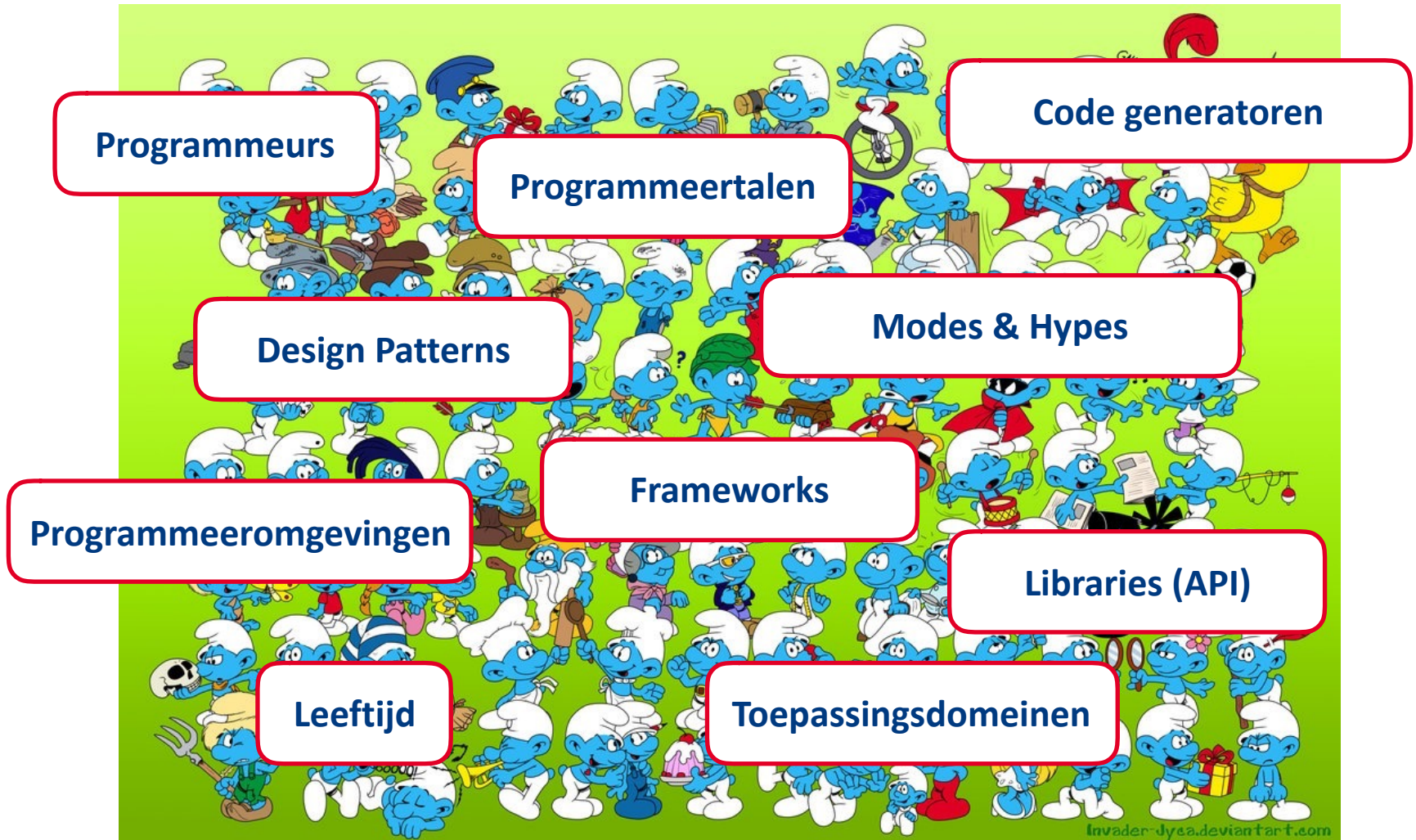
Google

2 miljard regels code



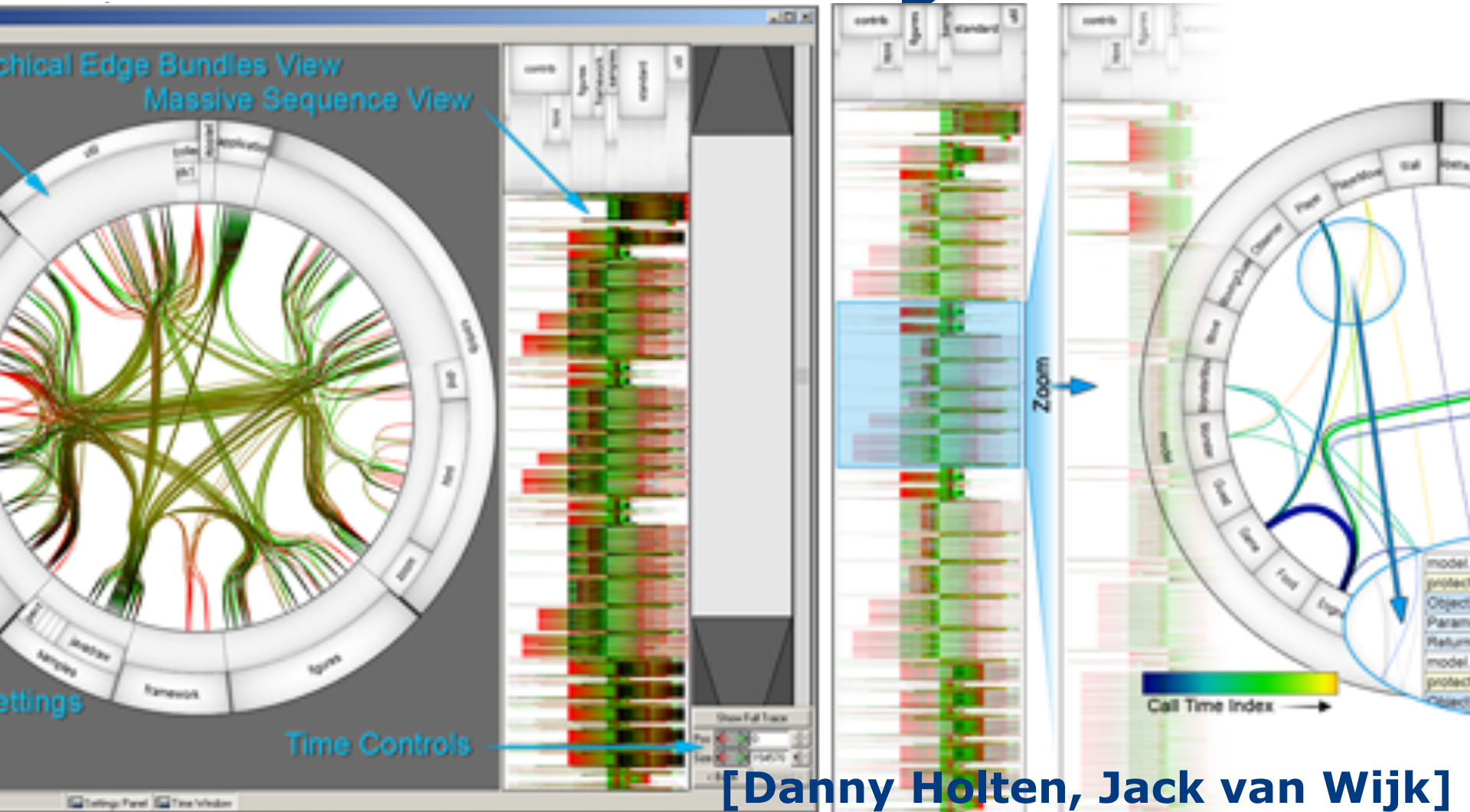
5000
70.000
1.000.000
20.000.000
2.000.000.000

Echte software is barok



Invader-dyca.deviantart.com

Echte software is ingewikkeld



[Danny Holten, Jack van Wijk]

Software uitersten



volume

**Gedetailleerd en
oppervlakkig
= Data**

**Gedetailleerd en
diep- of onzinnig
= Complex**

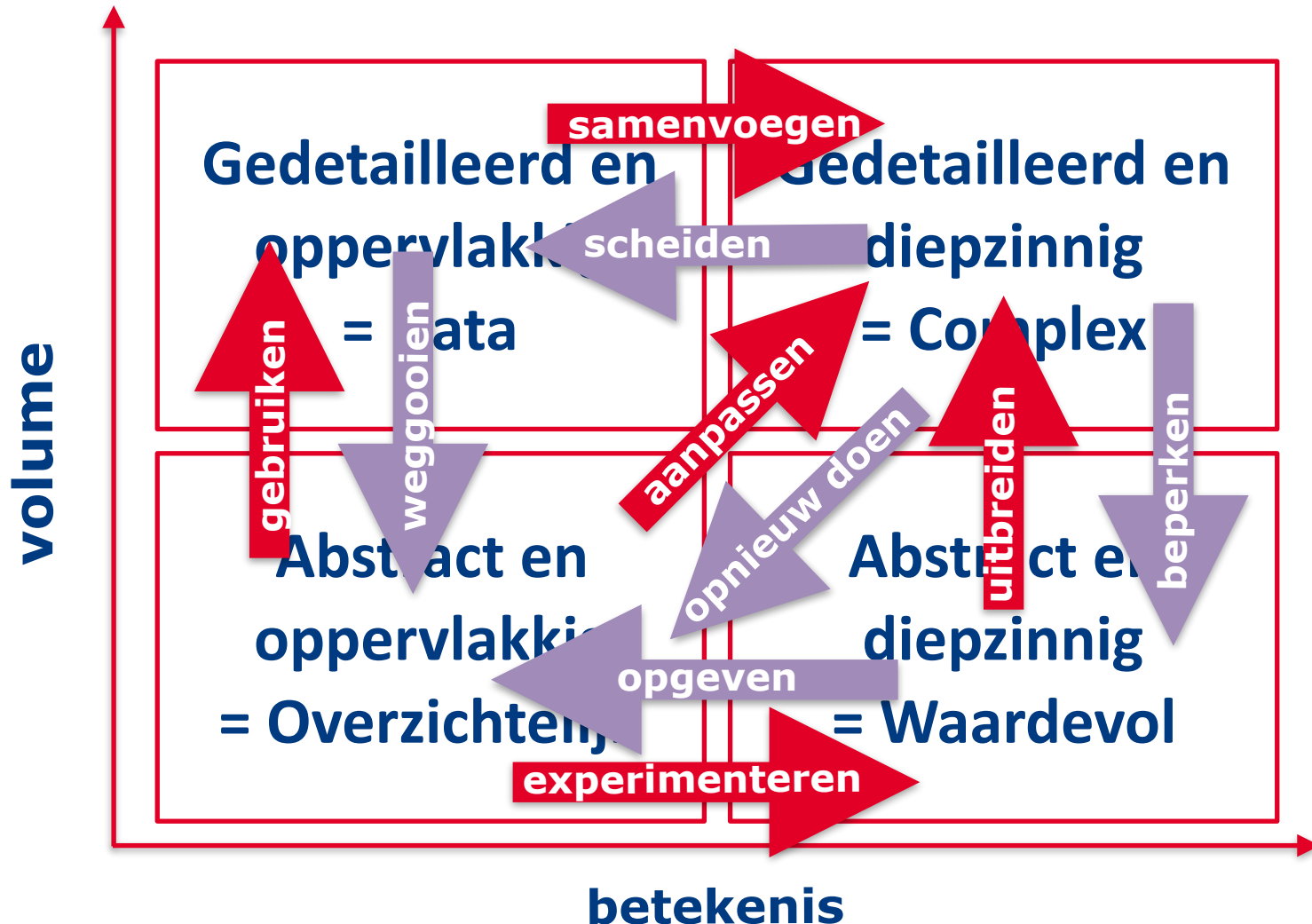
**Abstract en
oppervlakkig
= Overzichtelijk**

**Abstract en
diepzinnig
= Inzichtelijk**



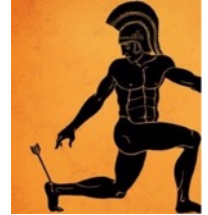
betekenis

Complexiteit lonkt





Waarom groeit software de pan uit?



Panta rei



Software moet
aangepast
worden

Software is te complex om
snel te begrijpen (of zelfs absurd)



Begrip is geen voorwaarde
voor arbitraire maar
"werkende" aanpassingen



groei van zowel broncode
als aantal verborgen
gebreken



Kortom

- Software is **fantastisch**
- Maar: **te complex** om te controleren en te garanderen
- Sluipende software problemen: **risicovol** vanwege impact
- De mens heeft hulp nodig:
 - De economie van goede analyse moet anders
 - Automatisch software analyse gereedschap



Intermezzo: voorkomen 1

- Even **niet** doen:
 - kerncentrale op internet aansluiten
 - internet + wifi-camera = baby monitor
 - CITO-resultaten op een web-server zetten
 - stemcomputers gebruiken
- Niet a priori al **grote software** gaan verzinnen
 - niemand zal weten wat het echt doet
 - niemand zal weten wanneer het af is
 - niemand zal weten hoeveel het echt gaat kosten
- Niet **betalen per uur** of **per regel**
 - daar krijg je meer broncode en complexiteit van
 - maar niet meer waarde of inzicht



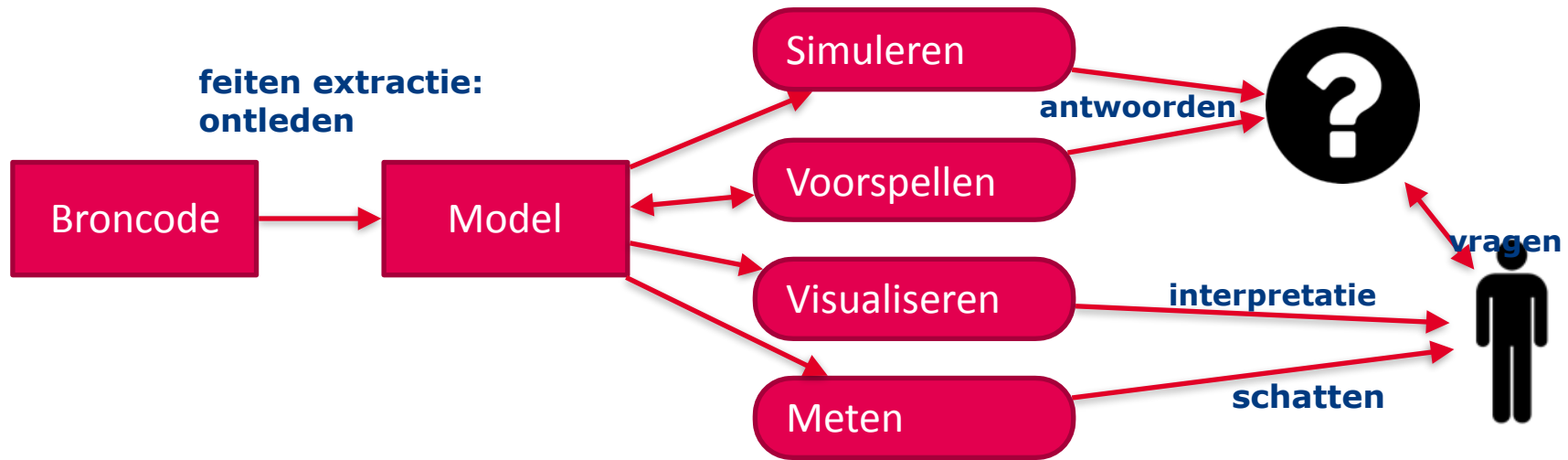
O TEMPORA O MORES

Intermezzo: voorkomen 2

- Heel anders aanpakken
 - Domein-specifieke programmeertalen
 - Model driven engineering
 - *Meer doen:* simuleren, verifiëren, genereren, ...
 - *met minder code.*
- Analyse nog steeds *noodzakelijk*
 - Modellen zijn ook broncode
 - Het platform kunnen begrijpen
 - Analyse voor **bestaande software** naar modellen
 - **Makkelijkere software wordt altijd meer software**
(denk aan de vurig verlangde herbruikbare componenten uit de jaren 90...)



Automatische software-analyse



**Computers
zijn (vaak) sneller,
nauwkeuriger
en vooral geduldiger.**

**en... al deze analyses
zijn slechts
benaderingen**

Analyse = vragen beantwoorden

- Waar zitten mogelijk fouten (“bugs”)?
- Welke onderdelen hangen van elkaar af?
- Waarom is deze software zo langzaam?
- Waarom gebruikt het zoveel geheugen?
- Wat moet ik controleren als ik dit verander?
- Kan dit vervangen worden door dat?
- Hoe worden onze prijzen berekend?
- Kan deze software per ongeluk iemand doden?
- Hoeveel gaat deze uitbreiding ons kosten?

waar zitten mogelijk fouten (“bugs”) in de broncode?

Wat is dan een bug?

Een **bug** is :

- een *effect* van software
- *niet de bedoeling*
- wel *gedefinieerd*
- en (*achteraf*) *geobserveerd*.



Bugs vooraf vinden is dus een vorm van de toekomst
voorspellen aan de hand van broncode analyse



Waar moet je dan overstappen?



"British Harrier accidentally bombs British carrier."



Error found in study of first ancient African genome
Finding that much of Africa has Eurasian ancestry was mistaken.

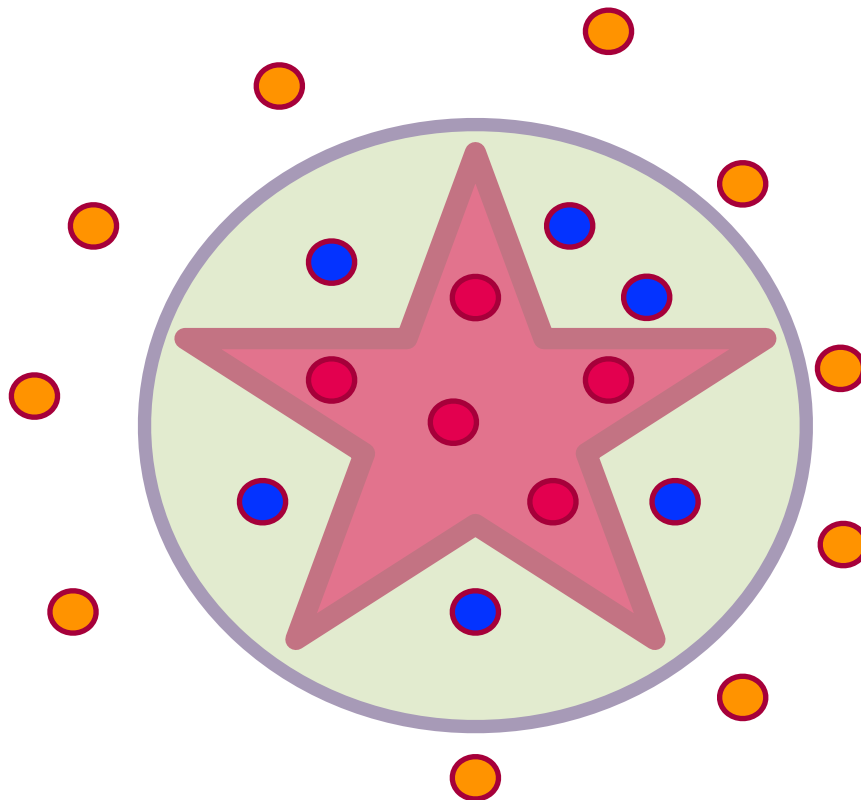


Internet Of Not-So-Smart Things: Samsung's Latest Smart Fridge Can Expose Your Gmail Password
(from the I'll-take-my-devices-stupid,-thanks dept)

<http://www.risks.org>

veilig overschatten:

geen gemiste bugs, maar veel vals alarm



échte oplossing



gevonden



goed gevonden



vals alarm



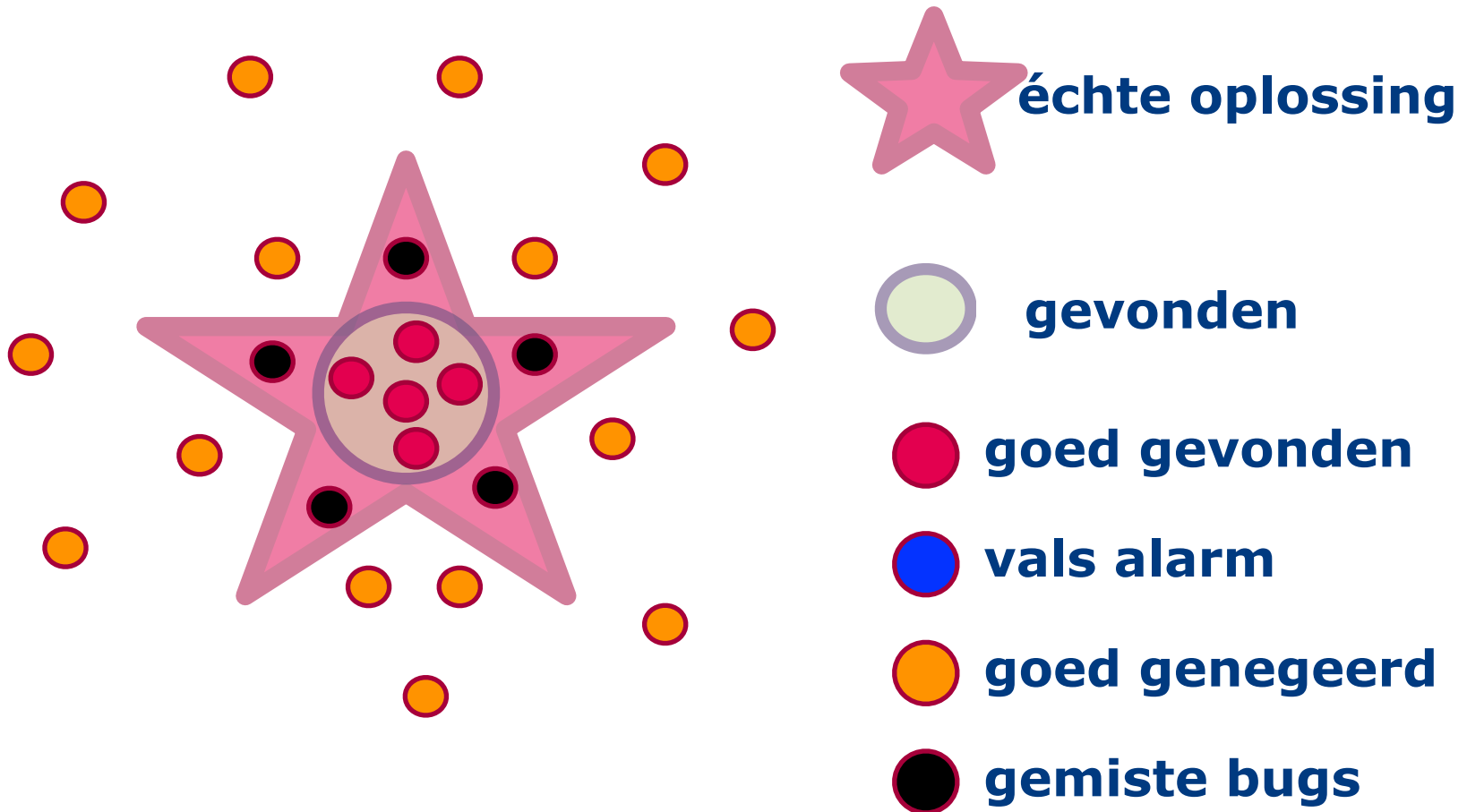
goed genegeerd



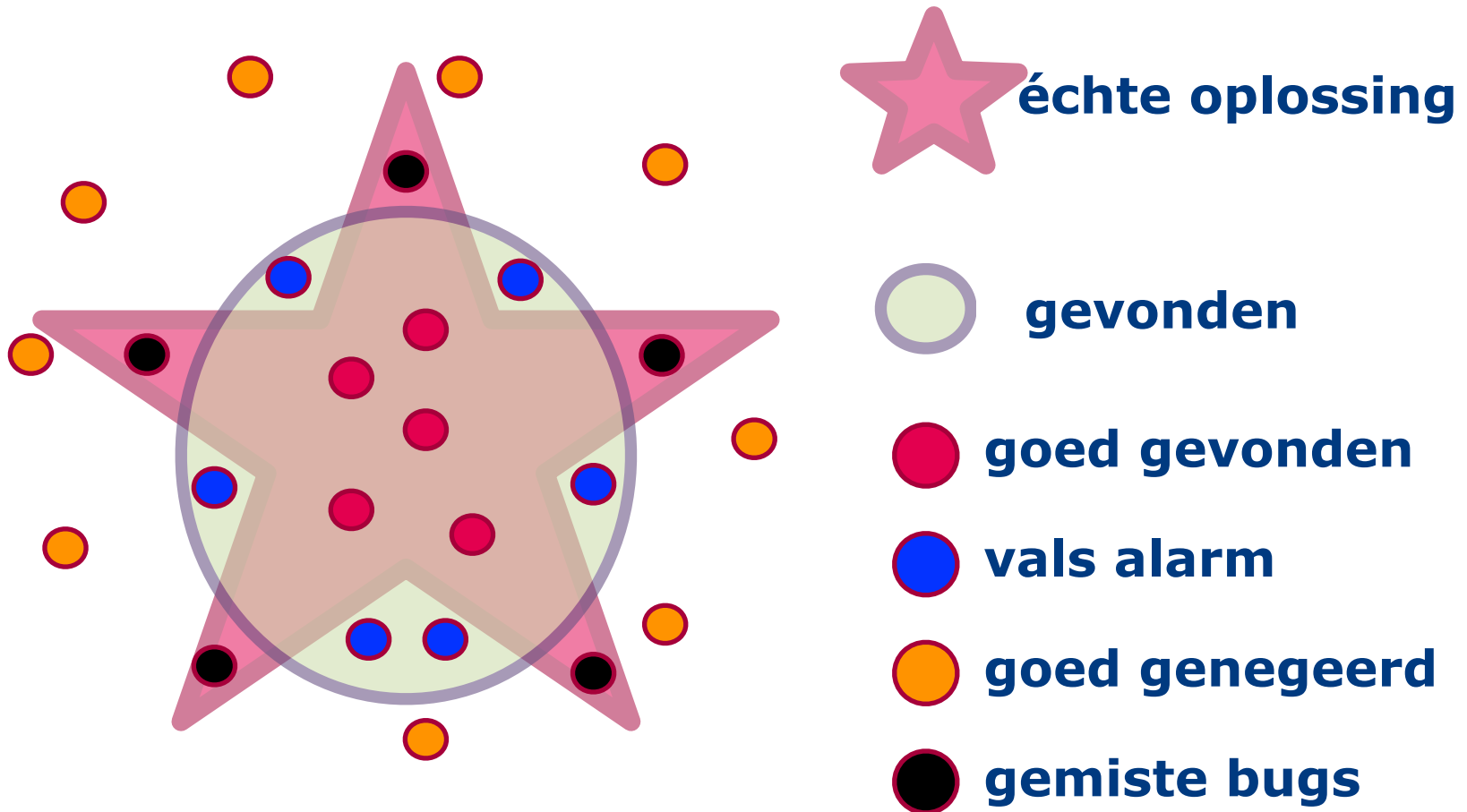
gemiste bugs

efficiënt onderschatten:

nooit vals alarm, maar je mist wel veel bugs



Onder- en overschatten tegelijkertijd: lijkt haast onvermijdbaar... maar creëert verwarring alom



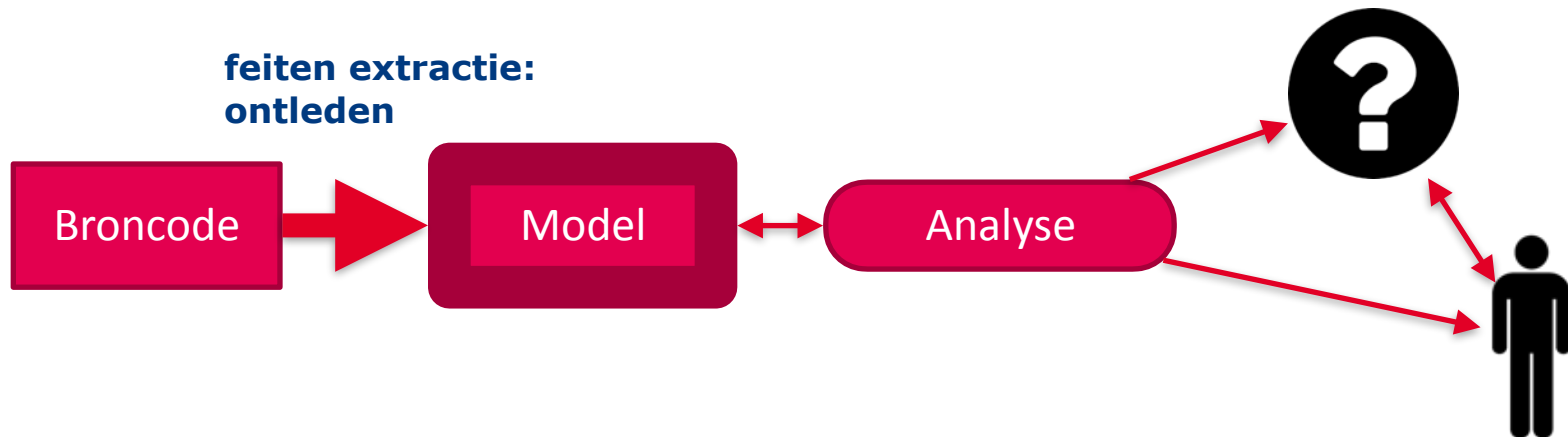


duur en traag
maar netjes



snel en goedkoop
maar slordig

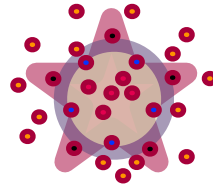
**Software analyses worden
nauwkeuriger (en duurder)
als we ze meer details voeden.**



Groter model kost onevenredig meer tijd en ruimte

Analysegereedschap evalueren

- **Empirisch onderzoek:** vergelijken van resultaten
 - t.o.v. van vorige versie
 - t.o.v. concurrerende aanpak
 - t.o.v. handmatige analyse
- **Demonstraties**
 - op zelf geselecteerd of geschreven code
 - proof-of-concept
- **Laboratoriumonderzoek**
 - op kleine groep projecten
 - controleren van variabelen
 - interne validiteit
- **Veldonderzoek**
 - op grote groep software projecten
 - externe validiteit



Analysegereedschap evalueren

- **Demonstraties**

- Inzichtig
- Maar niet overtuigend



- **Laboratorium:**

- reproduceerbaar, objectiverend
- selectief
- Niet te verwarren met externe validiteit
- Goede ontwikkeling: “artefact evaluation”



Kwaliteit van analysegereedschap

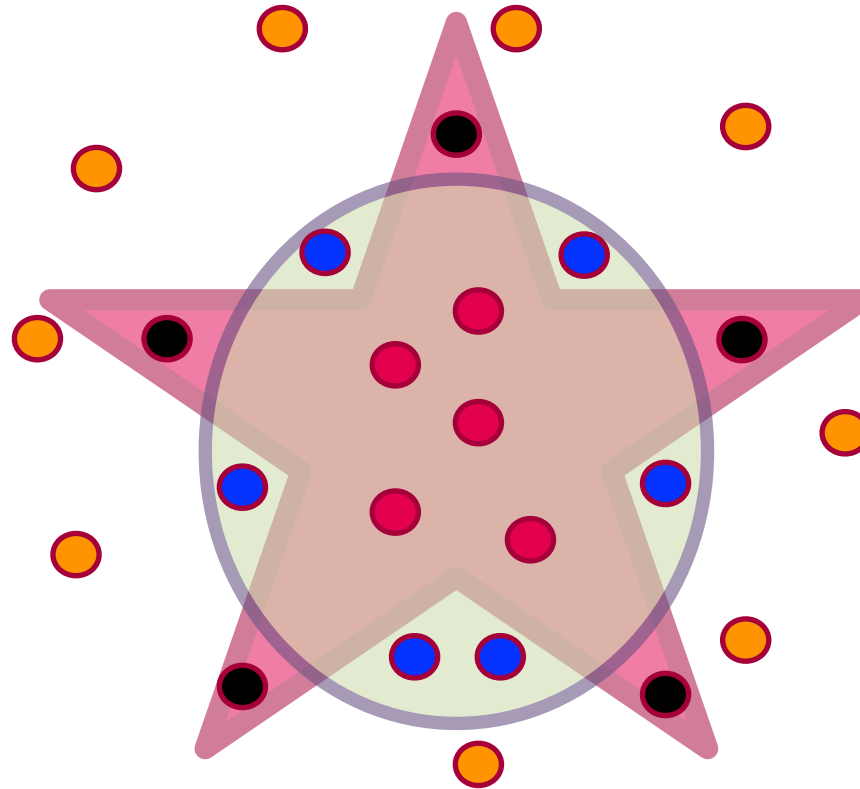
- **Veldonderzoek**
 - software repositories gebruiken
 - objectief en relevant
 - **verwarrend**: kwaliteit van gegevens
 - kip/ei: gebrek aan kennis over de data
- Niet te verwarren met interne validiteit
- Toekomst: samengaan van veld- en lab-onderzoek:
 - **benchmarking**
 - grote en goede **corpora**
 - met **bekende analyseresultaten**



**Zelf
software
analyses
ontwerpen
en dan
valideren
is
netelig**



[MC. Escher]



Catch 22:
Nauwkeurige analyses schalen niet
En snelle analyse zijn slordig

Software is domein-specifiek

- Alle software is gemaakt een context
 - en voortdurend mee-geëvolueerd
- Algemene analysetechnieken zijn beperkt tot algemene observaties...
- Software engineers gebruiken allerlei achtergrondkennis...
- Kunnen slimme automatische analyses dat niet ook?

Wanneer context helpt

dame?



jam!

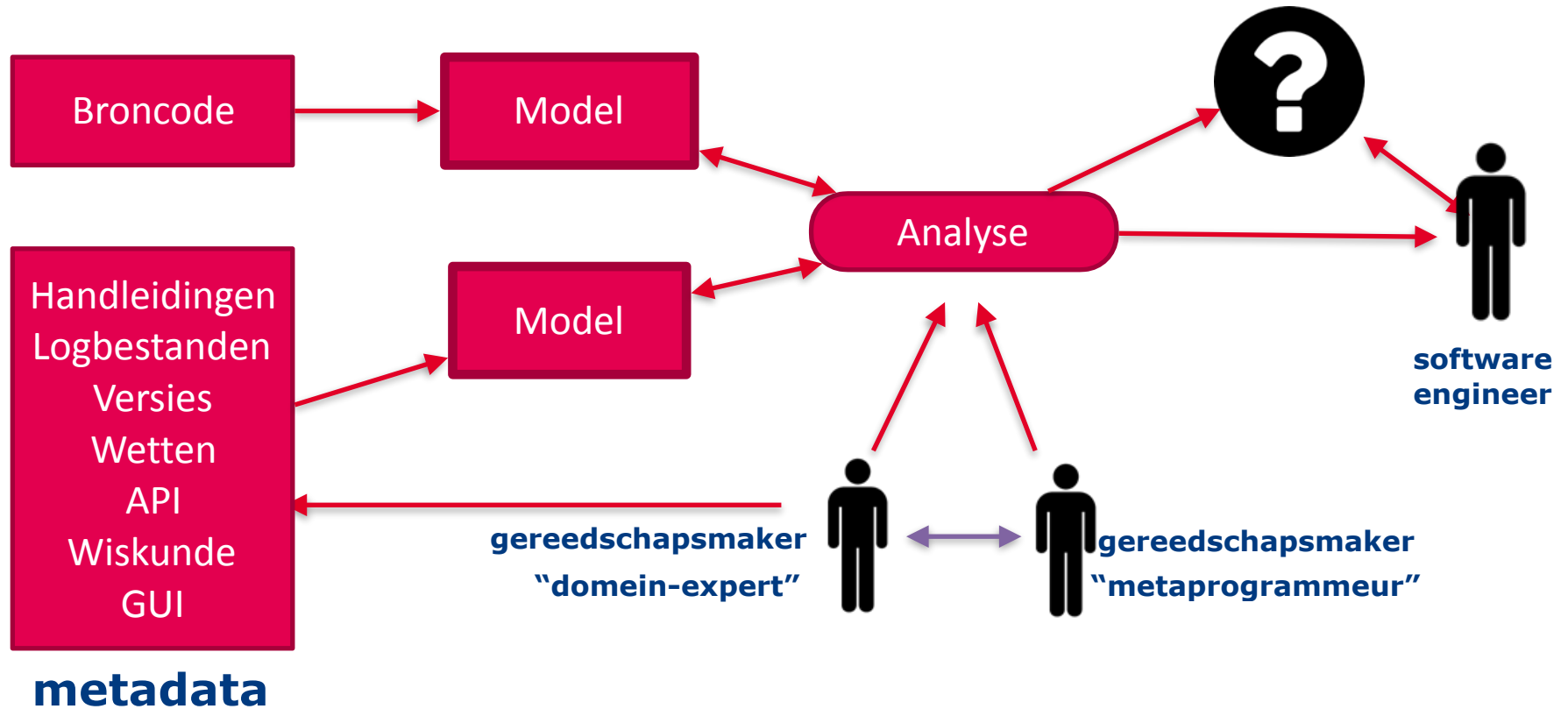
pindakaas!

[Vik Muniz]

Aanwezige context informatie

- Expertkennis (theorie over het domein)
- Redundante specificaties/modellen
- Broncode commentaar
- Test cases
- Pakket van eisen (requirements)
- Logbestanden
- Handleidingen (van frameworks)
- Labels in gebruikersinterfaces
- Vorige versies
- Foutmeldingen en foutrapportages

Samenwerken noodzakelijk



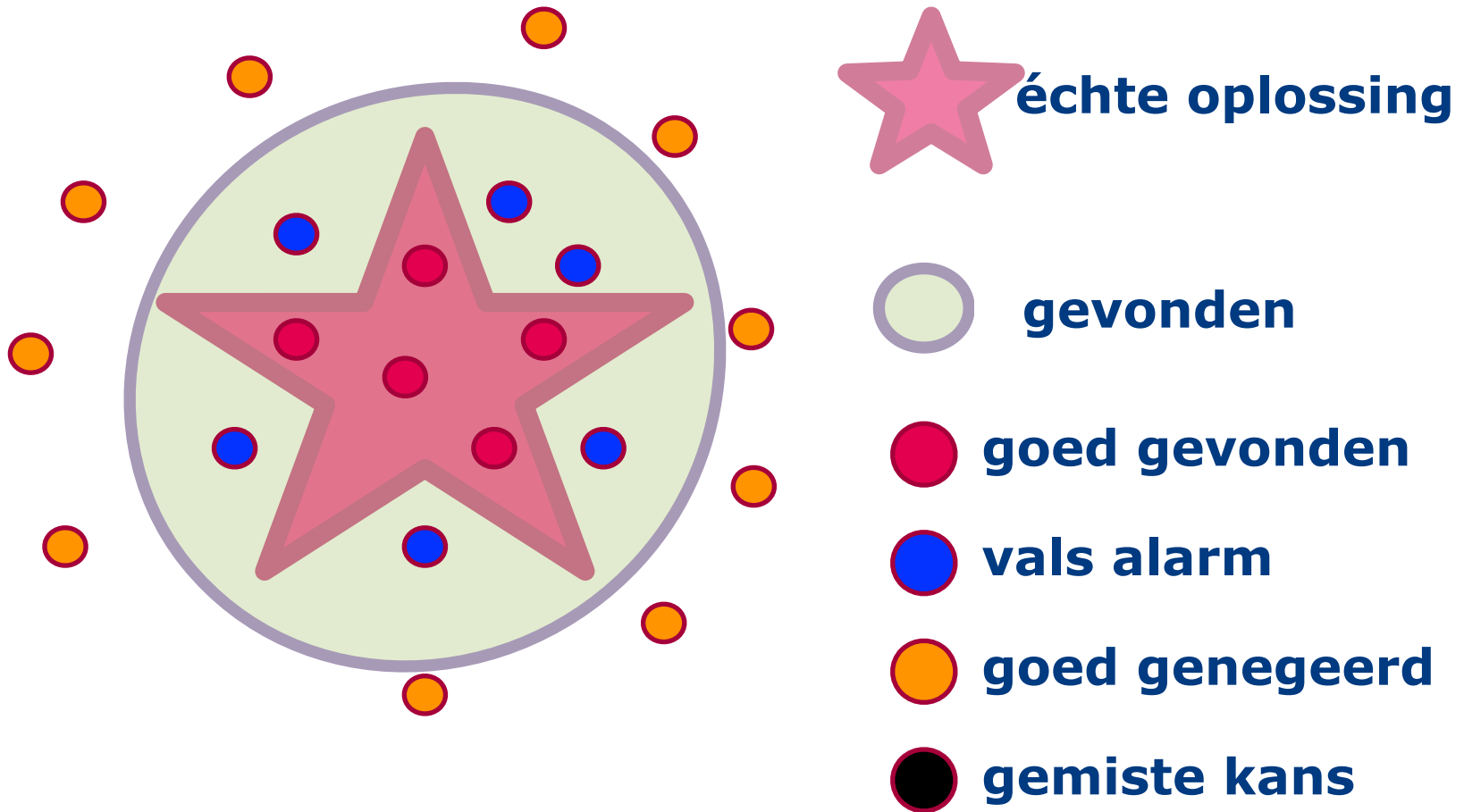
Context + Analyse = Frisse kijk

- + Broncode analyse
 - + Domeinkennis
 - + Meta-data
 - + Model driven engineering
-
- = Domein-specifieke software analyse

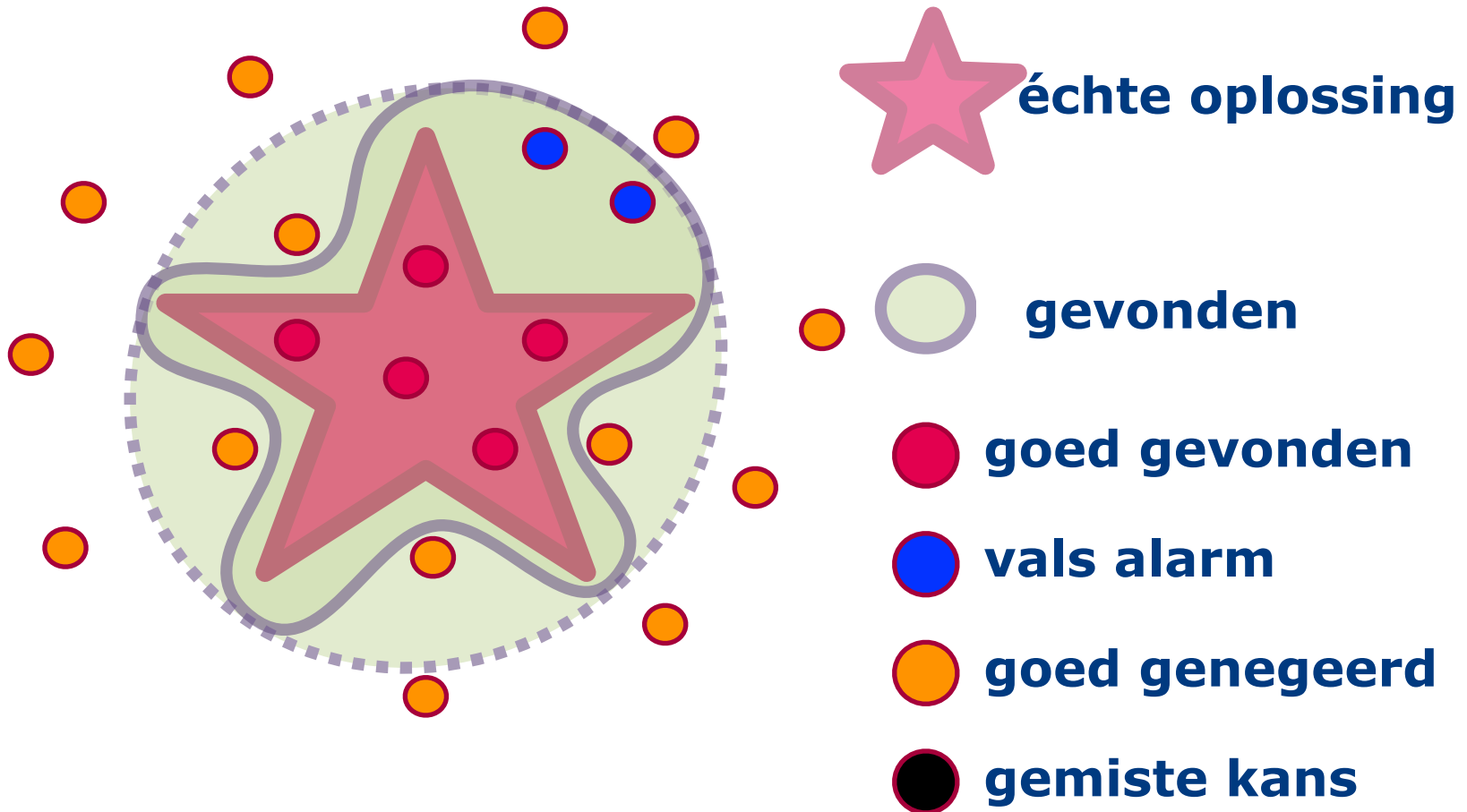
+ Teamwork MDSE sectie

+ Teamwork andere
onderzoekers en engineers

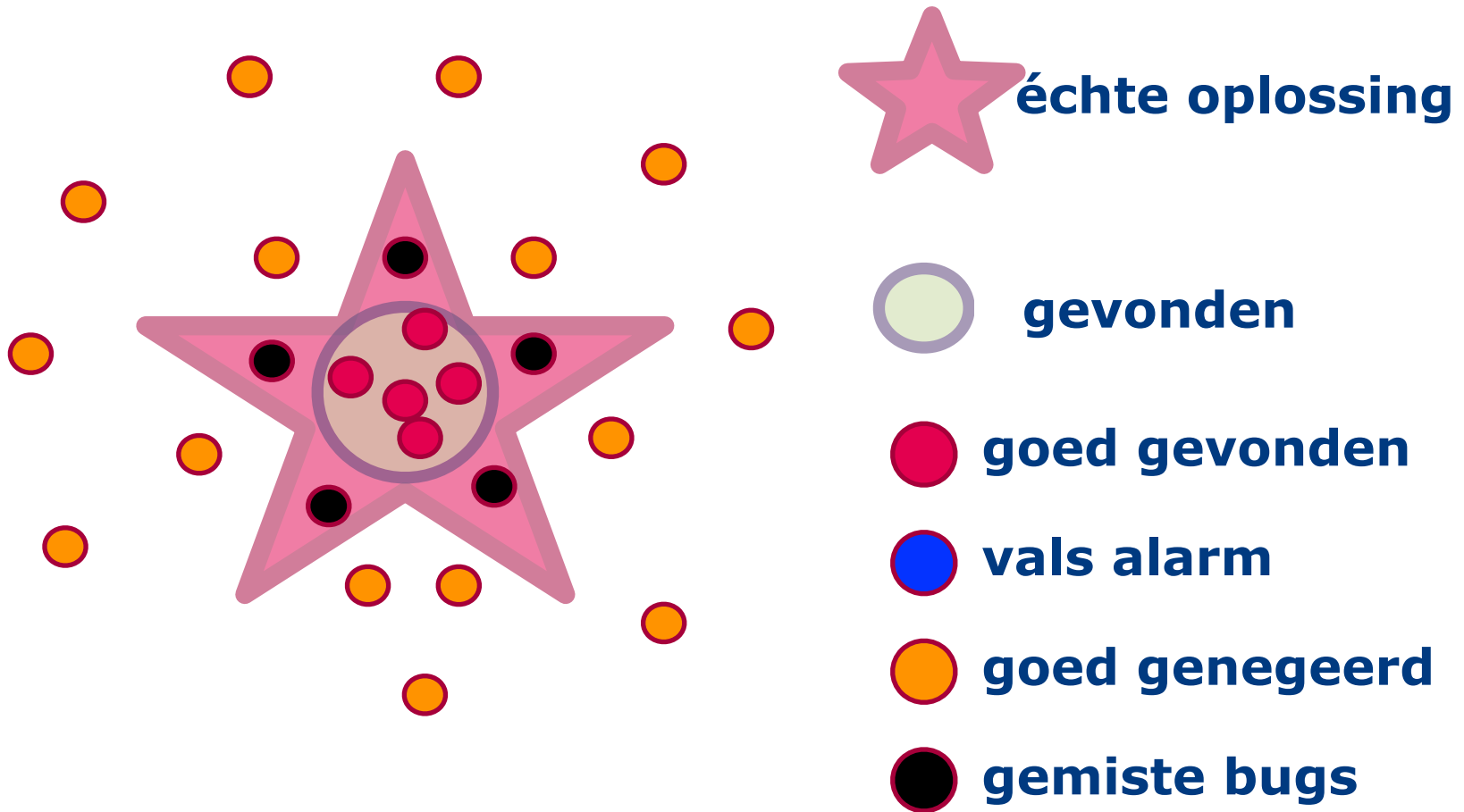
Domein-specifieke software analyse: Vals alarm wegsnijden met kennis



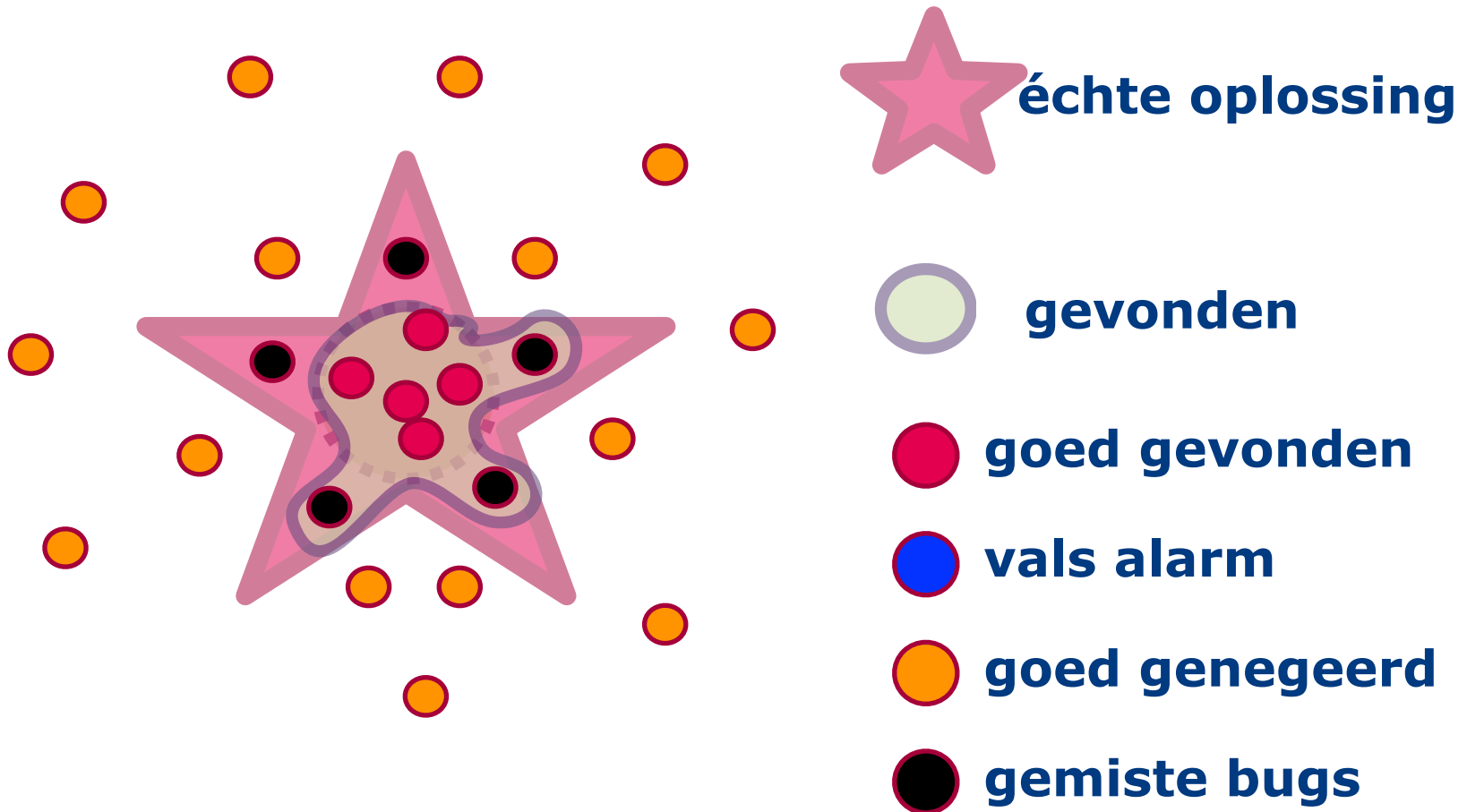
Domein-specifieke software analyse: Vals alarm wegsnijden met kennis



Domein-specifieke software analyse: oplossing groeien met kennis



Domein-specifieke software analyse: oplossing groeien met kennis



Domein-specifieke software analyse

- [illegible]

NWO “MERITS” project

- **Philips Healthcare**
- **Domein-specifieke model extractie:**
 - van broncode naar action models
 - waardevermeerdering van legacy software
- Ingrediënten
 - + programmeertaal-analyse met *Rascal*
 - + healthcare domeinkennis
 - + model-analyse met *mCRL*

- = model driven re-engineering



Gereedschap voor gereedschap

Broncode en meta-
data kunnen
ontleden om feiten
te extraheren

Patronen kunnen
vinden in broncode
en modellen van
broncode

Kunnen omgaan met
de variabiliteit van
broncode en meta-
data

Kunnen redeneren
over broncode
modellen

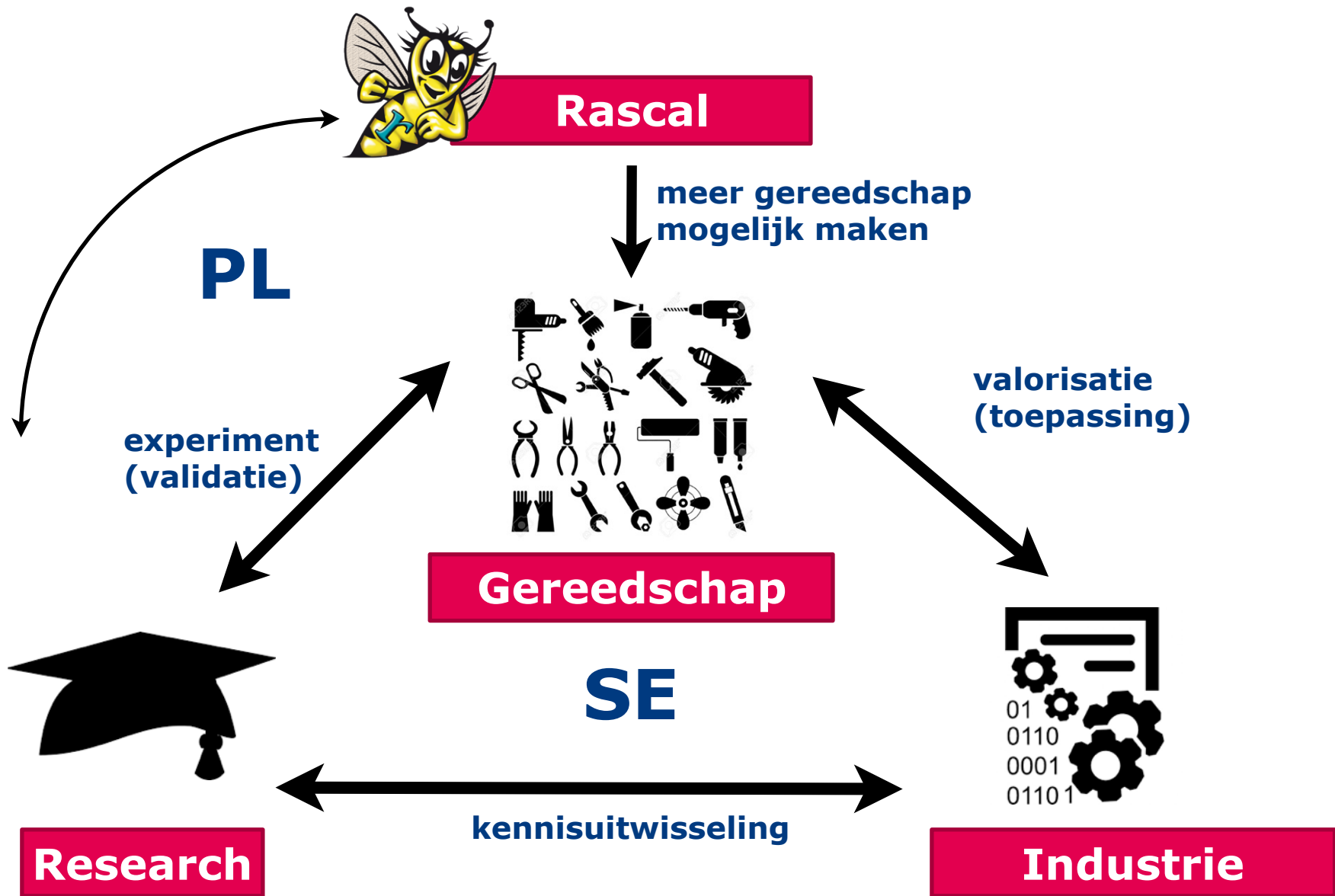
Inter-actief analyses
kunnen exploreren en
visueel rapporteren

Kunnen concentreren op de
essentie door gebruik van
krachtige operatoren en
herbruikbare bibliotheken



Rascal

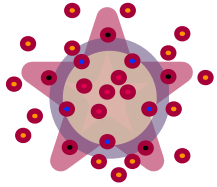
"one-stop-shop"





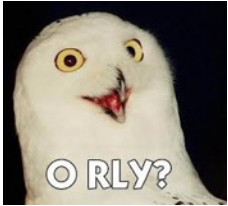
Onderzoeksvragen Metaprogrammeren

1. **Data acquisitie:** Hoe kan externe informatie over broncode effectief geïntegreerd worden in een grotere context van broncode-analyse?
2. **Variabiliteit:** Hoe kunnen we veel meer hergebruik van (complexe) analyses tussen programmeertalen en versies van programmeertalen bewerkstelligen?
3. **Schaalbaarheid:** Hoe kunnen we grotere hoeveelheden van gestructureerde informatie over broncode in één keer analyseren zodat kruisverbanden tussen verschillende onderdelen kunnen worden onderzocht?



Onderzoeksvragen Software Analyse

1. **Code-to-model:** Hoe doen we dat op een accurate manier?
2. **Context:** Hoe integreren we meta-informatie en configuratie-informatie over softwaresystemen in de statische analyse van broncode?
3. **Kwaliteit:** Hoe beoordelen we zo efficiënt mogelijk, (semi-automatisch) belangrijke kwaliteitseigenschappen zoals informatieveiligheid, beschikbaarheid en onderhoudbaarheid van software?
4. **Renovatie:** Hoe kunnen we voortdurende verandering en verbetering van softwaresystemen ondersteunen met behulp van de volgende generatie software analyses?



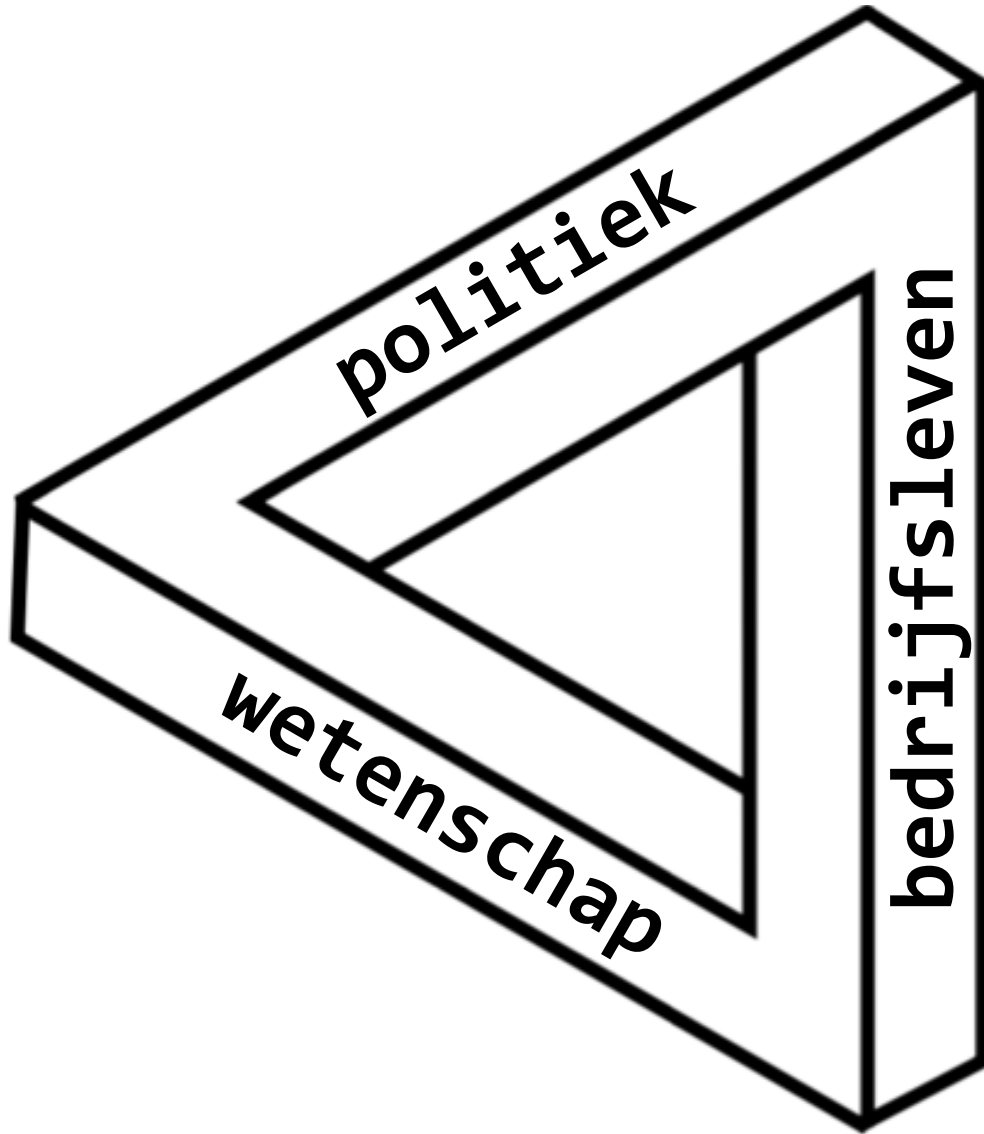
Onderzoeksvragen Empirische Valuatie

1. **Wat is de nauwkeurigheid** van nieuw en bestaand gereedschap en wat kan de (positieve of negatieve) **impact** zijn van het gebruik van dit gereedschap op de broncode en het ontwikkelingsproces?
2. **Wat is 'normale' broncode?** Wat mogen we verwachten van de impact van nieuw gereedschap op de kwaliteit van bestaande broncode en het ontwikkelingsproces ten opzichte van de ruis van andere factoren?
3. **Wat is de validiteit van meta-informatie** (bijvoorbeeld uit versiebeheer-systemen en issue-trackers) die gebruikt wordt bij het evalueren van de effectiviteit van gereedschap voor software-analyse?



Onderzoeksvragen Valorisatie

1. Welke uitdagingen met betrekking tot software-analyse hebben nu en straks de hoogste prioriteit?
2. Hoe creëren we eigenaarschap van maatwerk-gereedschap voor software-analyse?
 1. Wat zijn noodzakelijke eigenschappen van het meta-gereedschap om dit te bewerkstelligen?
 2. Welke opleidingsmiddelen zetten we hierbij in?
 3. Welke investeringen in gemeenschapsvorming zijn nodig?
3. Hoe benaderen we de implementatie van nieuw softwaregereedschap in complexe bedrijfscontexten?



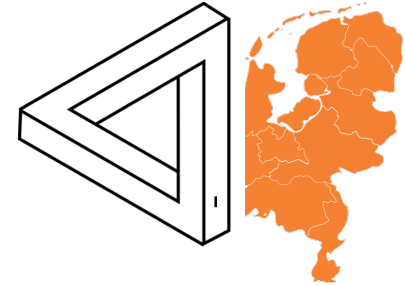
**samenwerken
is noodzaak**

vooral op korte
termijn

**samenvoegen
is absurd**

lange termijn
moet onafhankelijk
zijn

Méér Nationale Samenwerking Software



- **VER**eniging **S**oftware **E**ngineering **N**ederland
Wordt lid via <http://www.versen.nl>



- Legacy Coalitie
 - Contacteer legacycoalitie@nwo.nl
 - of Ad Kroft ad.kroft@inseit.nl



Wist u dat?



- We met zijn allen onlangs de Vereniging Software Engineering Nederland (VERSEN) hebben opgericht?
 - Wordt lid via <http://www.versen.nl>
- We met zijn allen onderzoeken hoe we samen de schouders onder de uitdagingen van legacy software kunnen zetten?
 - Contacteer legacycoalitie@nwo.nl
 - of Ad Kroft ad.kroft@inseit.nl

**College “Inleiding
Automatische
Software Analyse”**

familie &
vrienden

**Agenda “kansen en
uitdagingen in
software analyse”**

collega’s software
engineering

of je hebt zo’n dag...

**Gewoon even
wachten op de
borrel**

collega’s uit
industrie, onderwijs & onderzoek

Uitnodiging tot samenwerken

politici,
beleidsmakers,
entrepreneurs

**rationeler
omgaan met
software**

geïnteresseerde master studenten
promovendi, post-docs, tenure trackers

Promotie veld software analyse

Menu

Om mee naar huis te nemen



- **Software is fantastisch**
 - Software is een innovatiemotor
 - Broncode beter leren begrijpen is geen luxe maar noodzaak
 - Software kansen wegen op tegen risico's; mits we...
- **Software kunnen analyseren**
 - Bovenmenselijk complex; dus kan alleen *automatisch*
 - *Context-informatie* is de sleutel
 - *Samenwerken* is essentieel
- **Meta-software-analyse**
 - Snel en goed integreren, specialiseren en uitbreiden
 - Experimenteren met méér analyse-gereedschap
 - <http://www.rascal-mpl.org>
- **Vacatures** PhD kandidaten en Post-doc