Architecten in Control met Principle Mining

14 november 2107

W.T. (Wim) Goes | Directeur Valori Software Improvement
VALORI | ORTELIUSLAAN 1000 | 3528 BD UTRECHT NL
T +31 30 711 11 11 | F +31 30 711 11 12 | M +31 6 5 0 99 9 66 8
E WIMGOES@VALORI.NL | WWW.VALORI.NL

Agenda

1. Introductie
2. De staat van softwareontwikkeling
3. Is de Architect nog wel in control?
4. Principes en Abstractieniveaus
5. Modelleren van Domeinkennis en Principes
6. Werkend maken in de Praktijk – De architect in control

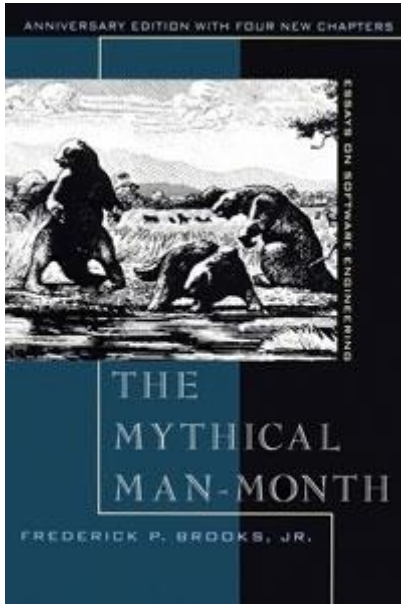
1. Introductie

- Ervaren IT Adviseur en Software Onderzoeker
- Directeur Valori Software Improvement
- Opleidingen
  - TU Delft Informatica
  - Nijenrode General Management
- Ervaring
  - Software Ontwikkelaar en Architect
  - All Round IT Adviseur
  - Software Onderzoeker

2. De Staat van Softwareontwikkeling (1)

Brooks Law: There is no silver bullet

"there is no single development, in either technology or management technique, which by itself promises even one order of magnitude improvement within a decade in productivity, in reliability, in simplicity."



	Ontwikkeling			Onderhoud	
Omvang (functiepunten)	Doorlooptijd (maanden)	Kosten (in miljoen Euro)	Kans op mislukken	Onderhoud (in fte per jaar)	Verwachte levensduur (jaren)
100 - 500	6 - 11	0,07 - 0,6	5% - 12%	0,1 - 0,7	3 - 5
500 - 1.000	11 - 15	0,6 - 1,6	12% - 17%	0,7 - 1,3	5 - 6
1.000 - 5.000	15 - 28	1,6 - 15	17% - 32%	1,3 - 7	6 - 8
5.000 - 10.000	28 - 36	15 - 40	32% - 39%	7 - 13	8 - 10
10.000 - ...	36 - ...	40 - ...	39% - ...	13 - ...	10 - ...

Ontwikkeling (20%)				Onderhoud (80%)		
Eisen 20%	Ontwerp 20%	Bouw 20%	Foutherstel 40%	Uitbreiding 60%	Aanpassing 20%	Foutherstel 20%

Bronnen

- Capers Jones, Software Assessments, Benchmarks and Best Practices, Addison-Wesley 2000
- Chris Verhoef, Quantitative IT Portfolio management, Science of Computer Programming 45, 2002
- Robert Glass, Facts and Fallacies of Software Engineering, Addison-Wesley, 2003

2. De Staat van Softwareontwikkeling (2)

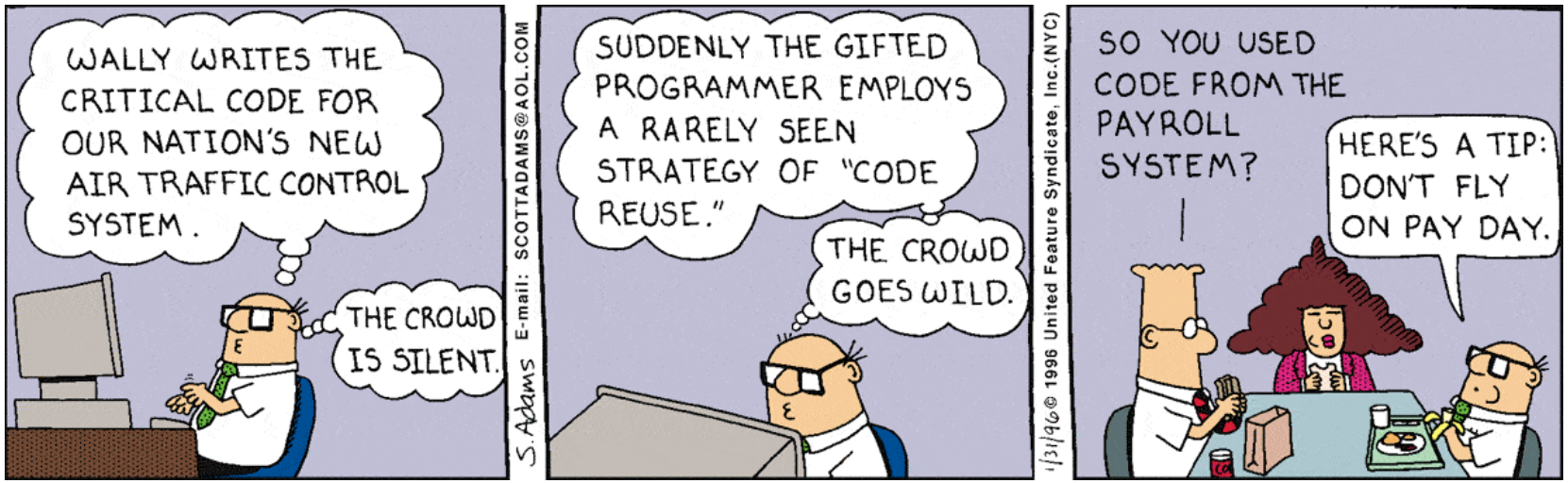
- Elke vijf jaar verdubbelt het aantal ontwikkelaars in de wereld*
- De eerste ontwikkelaar: Alan Turing (1945)

We shall need a great number of Mathematicians of ability, because there will be a good deal of work of this kind to be done.

One of our difficulties will be the maintenance of an appropriate discipline, so that we don't loose track of what we are doing

*The future of programming (Youtube, "Uncle" Bob Martin - "The Future of Programming")

2. Is de Architect nog wel in control?



4. Principes en Abstractieniveaus (1)

- Begripsvorming op verschillende niveaus van abstractie
  - The purpose of abstraction is not to be vague, but to create a new semantic level in which one can be absolutely precise (Edsger Dijkstra, The Humble Programmer, 1972)
  - The entire history of software engineering is that of the rise in levels of abstraction (Grady Booch, The Promise The Limits The Beauty of Software, 2007)

4. Principes en Abstractieniveaus (2)

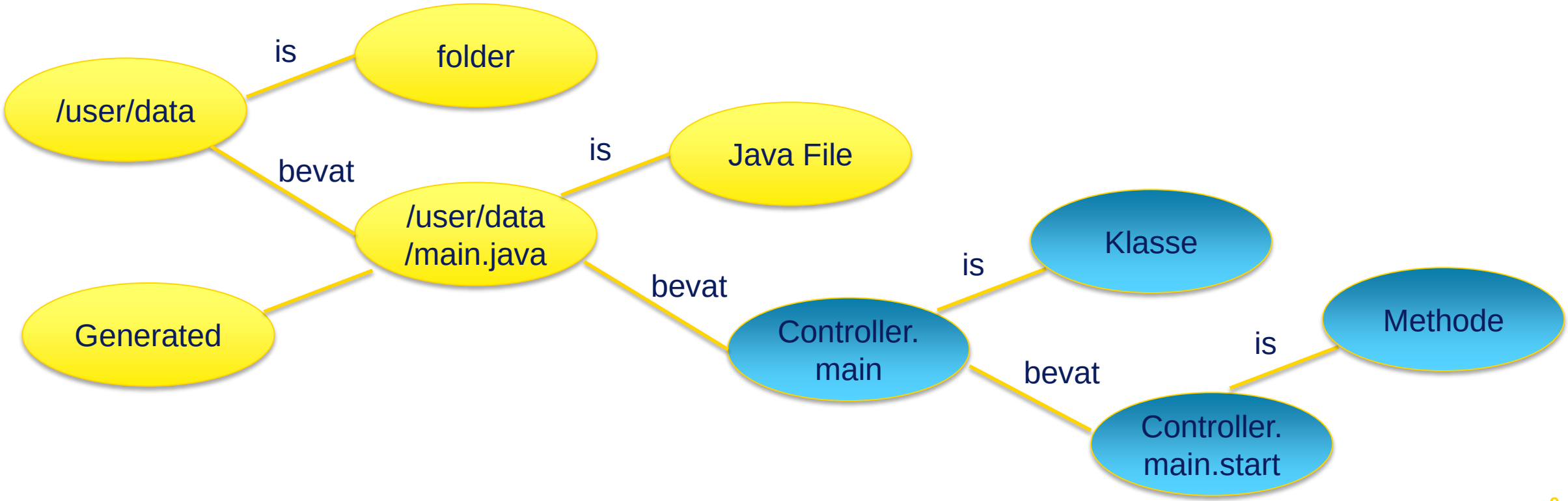
- Principes behoren bij een bepaald abstractieniveau of view

Abstractie	Principes
Business Services	Bewerkingen op bepaalde gegevens, hergebruik
Lagen van de applicatie	Functies van lagen / delen en relaties
Klassen of modules	Separation of concerns, cohesie en coupling
Variabele- en functienamen	Naamgeving van variabelen en functies

- Principes
  - Voor gebruik op het niveau van de abstractie
  - Voor gebruik door een hoger abstractieniveau
    - Mappen van niveaus (reflection)
    - ‘lekken’ zoveel mogelijk te voorkomen (Law of Leaky Abstractions)

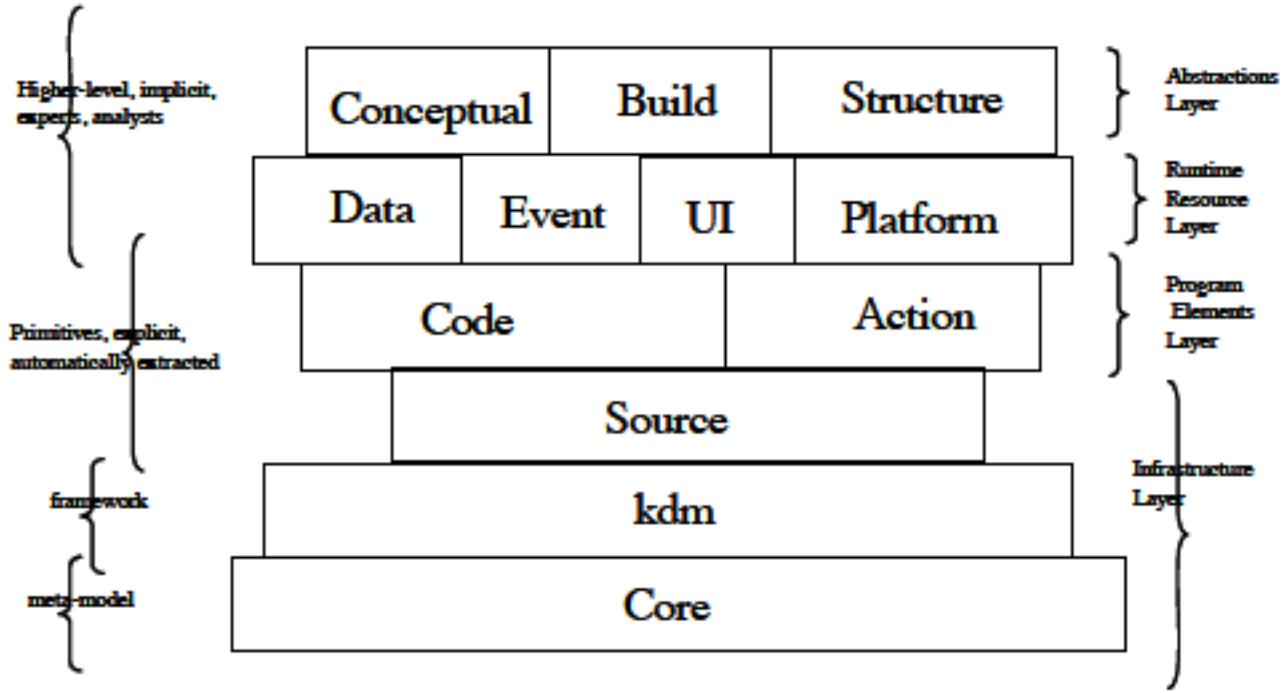
5. Modelleren Domeinkennis en Principes (1)

- De basis voor een expertsysteem is het vastleggen van de kennis over de objecten van onderzoek. Kennis kan vastgelegd worden op basis van zogenoemde linked data. Deze linked data geven de semantische relaties tussen elementen weer.

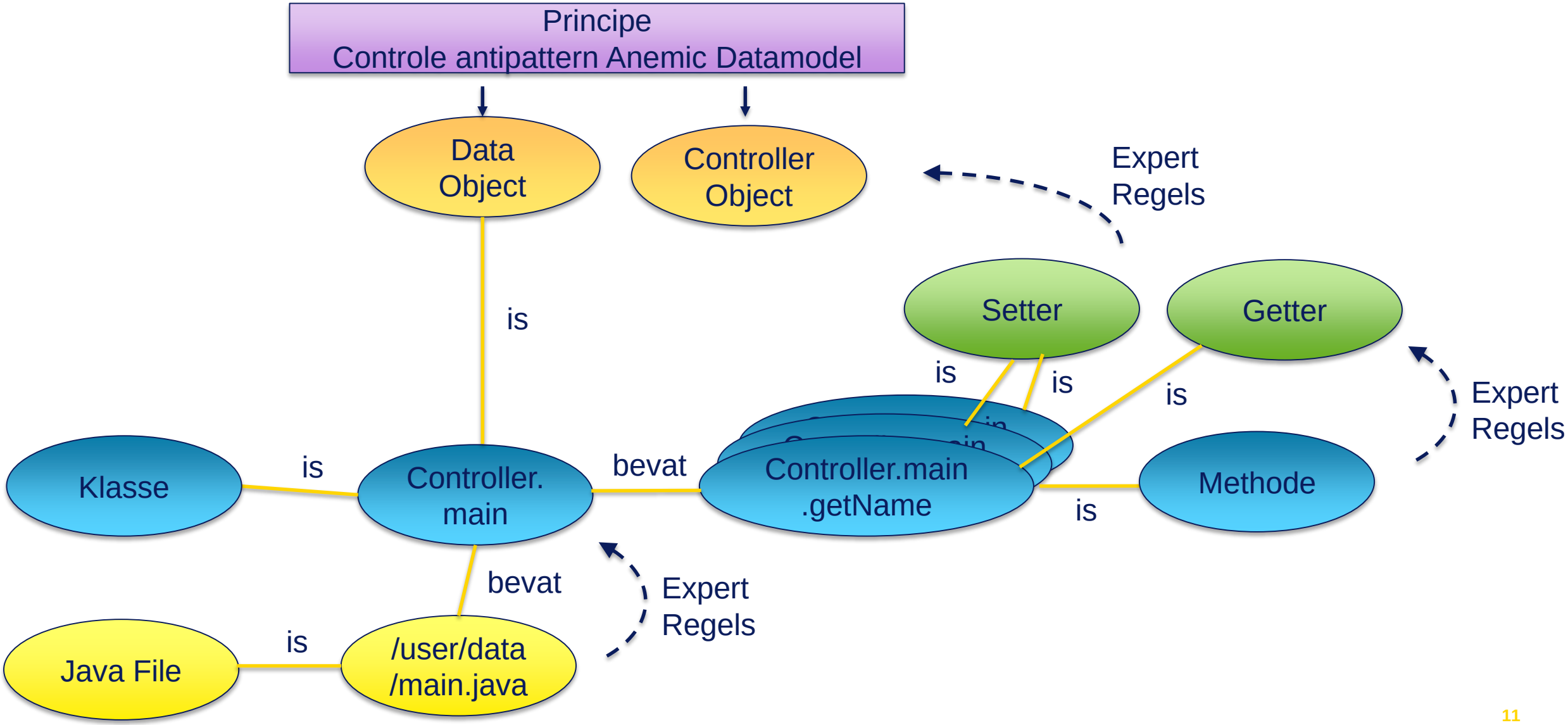


5. Modelleren Domeinkennis en Principes (2)

- Object Management Group (OMG) heeft een specificatie uitgebracht die als basis kan dienen voor het modelleren van domeinkennis en waarop voortgebouwd kan worden. Dit is het zogenoemde Knowledge Discovery Model (KDM)

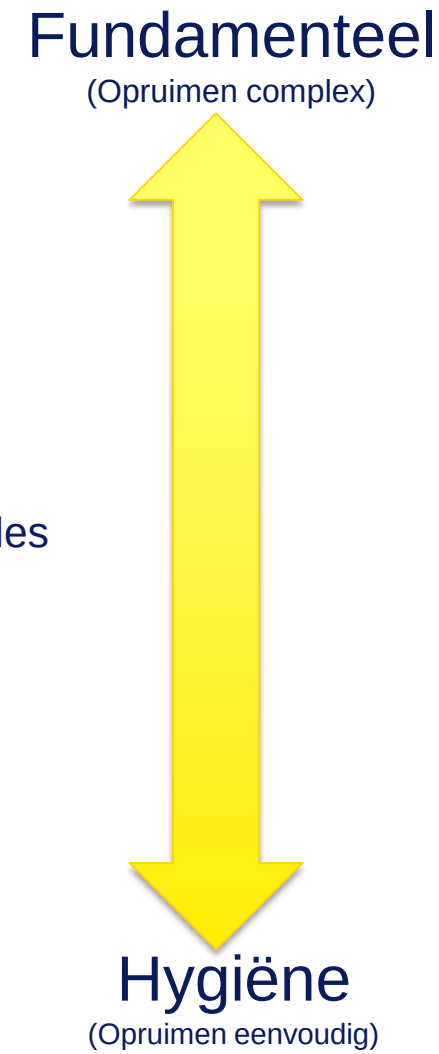


5. Modelleren Domeinkennis en Principes (3)



6. Werkend krijgen in de Praktijk
De architect weer in control (1)

- Softwareconstructie op de verschillende niveaus
  - Ontwerp- en architectuurpatronen
    - Object Oriented ontwerppatronen
    - Regels over aanroeprelaties
    - Project of systeemspecifieke regels
  - Unit niveau patronen
    - Fout- en exceptie-afhandeling
    - Dataflow patronen, bijvoorbeeld security controles
    - Stringmanipulatie in relatie tot performance
    - Unitomvang en -complexiteit
  - Code statement level patronen
    - Regellengte
    - Conditiecomplexiteit
    - Naamgeving van variabelen



6. Werkend krijgen in de Praktijk
De architect weer in control (2)

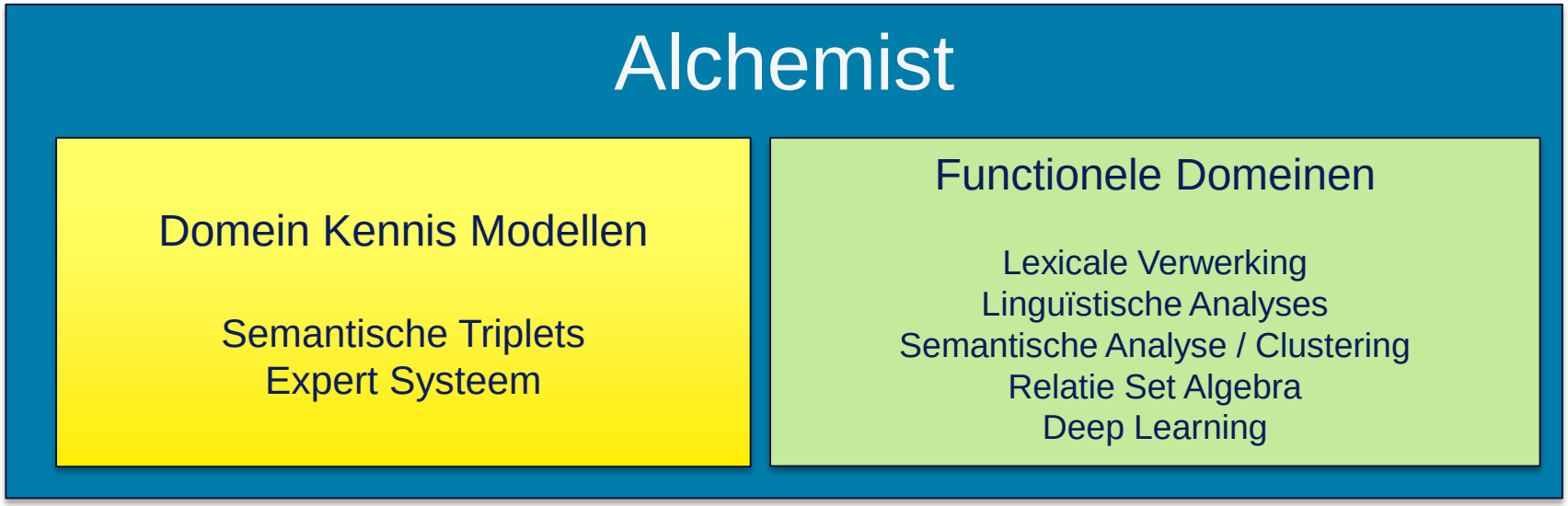
Toegevoegde Waarde

Rapportages
(Voortgang en Kwaliteit)

Inzicht
(Business & Development)

Verbeteringen
(onderhoud)

Transformatie
(Modellen, VDML)



6. Werkend krijgen in de Praktijk
De architect weer in control (3)

Define

Definieer Principes
- Verschillende niveaus
- Generiek en Specifiek

1-to-1 Translation

Automatische Controles
- Generiek en Specifiek
- Detail en Architecture
- Principes

Feedback

Kortom:

Ik maak me zorgen over de **verloedering van software-ontwikkeling** in een wereld die in toenemende mate software-afhankelijk wordt

Ordening en **discipline** zijn de pijlers van ons vak. Zonder **gesloten feedback-lus** komen we echter nooit ‘in control’.

Principle mining kan en moet. Doet u mee?

Wim Goes
Valori Software Improvement
E: WimGoes@valori.nl
T: 06 50 999 666