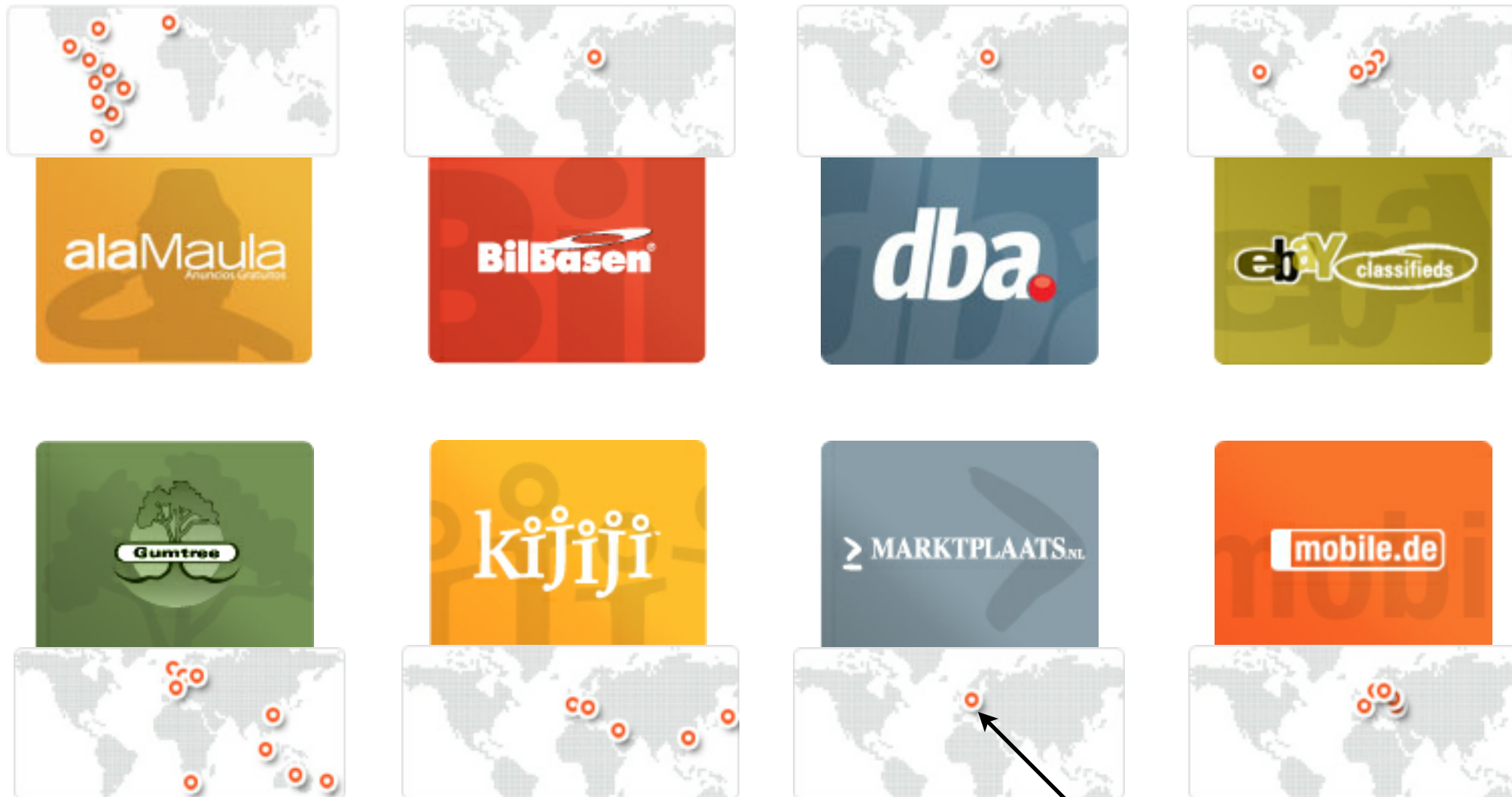


Anatomy of a Snowflake

Bartosz Kiera
Goto Amsterdam 2012



you are here



My road to Agile



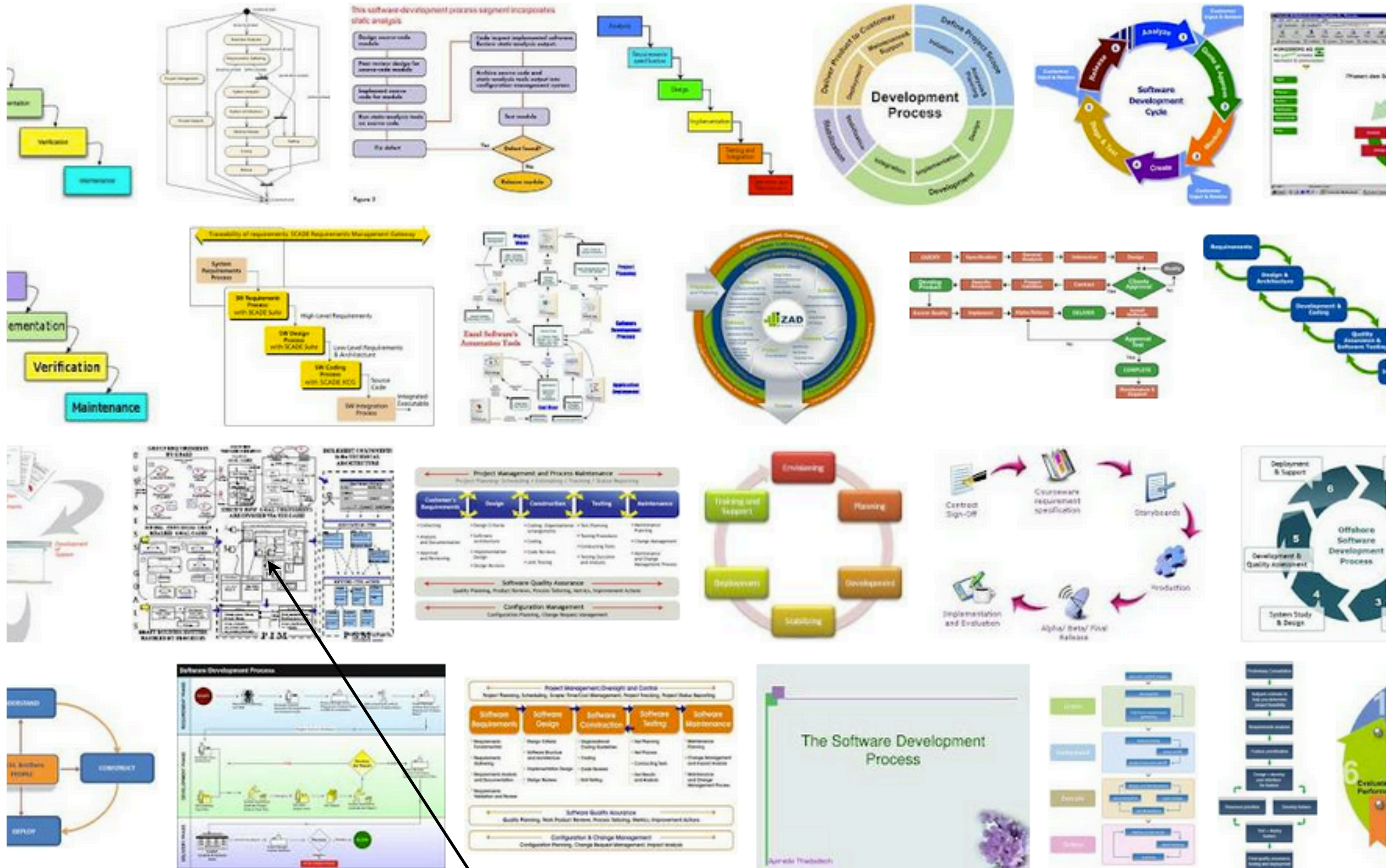
Patterns are evil



The Fantastic 4



My road to Agile



yes, that's a process

Understanding RUP roles

[Anthony Crain](#), Software Engineering Specialist, IBM, Software Group

Summary: from The Rational Edge: This article provides a simple way to conceptualize and recall the numerous roles defined in the IBM Rational Unified Process, or RUP. In addition, it suggests how to match role assignments to personality type, using the Myers-Briggs Type Indicator.

Tags for this article: [rup](#)

 [Tag this!](#)  [Update My dW Interests](#) ([Log in](#) | [What's this?](#))

Do you think IBM Rational Unified Process,® or RUP,® defines more roles than you can remember? Read on! You'll discover a hidden pattern that will make remembering them a cinch. You'll also find guidance for making successful role assignments, based on your team members' personality types.

now I'm convinced...

NTM - NOTAM Management

IFS Briefer - Globe

FOS - *Flight Operation Suite*

BRF - Briefer

ULD - ULD Control

WXS - Weather Station

QUE - *Communication & Alerting*

FIDS - Flight Information Display System

LST - Weight & Balance

FIDS - Flight Information Display System

CDM - Crew DX

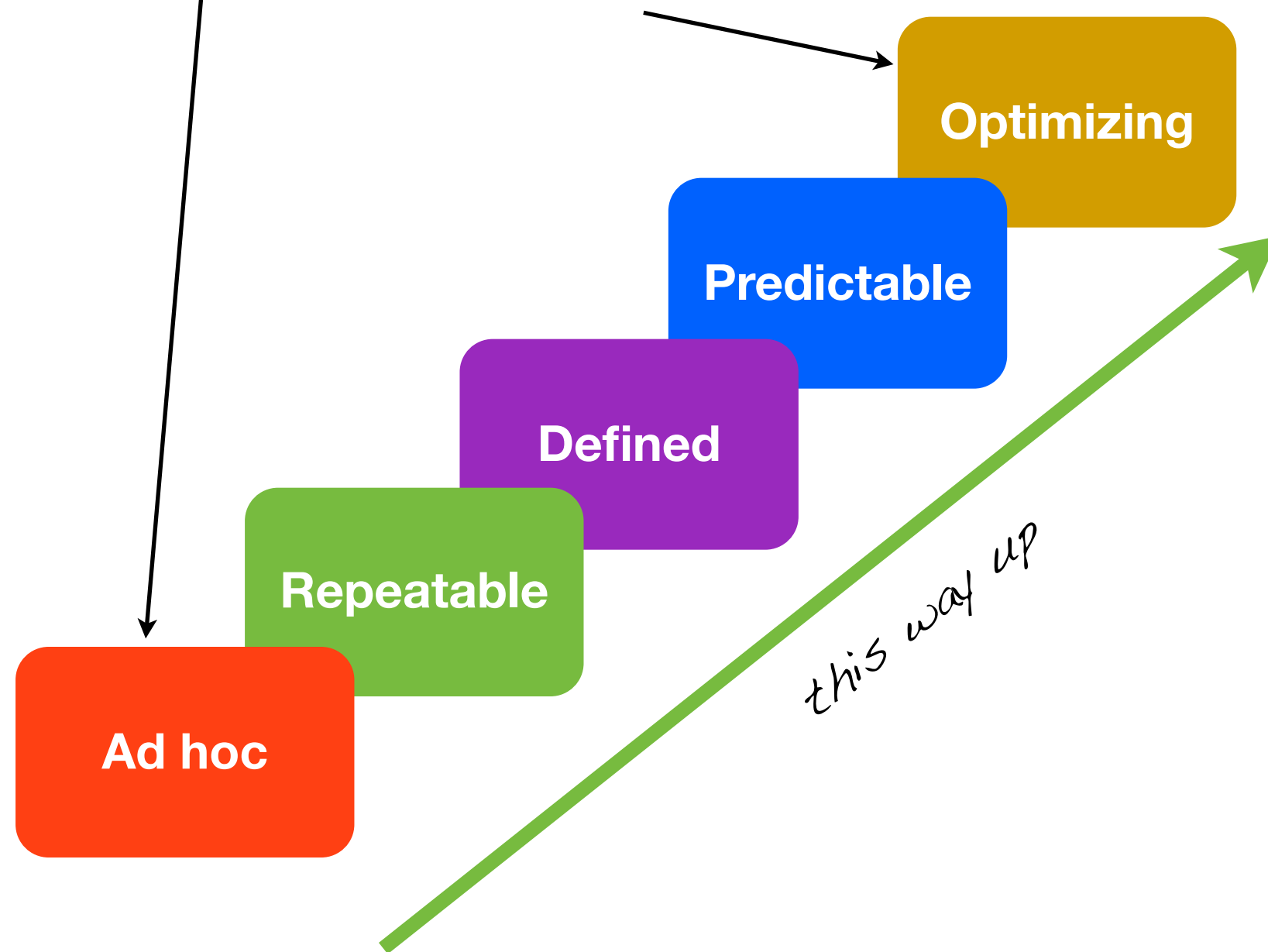
FLW - Flight Watch

TOP - Aircraft Performance Tool

6 developers

2 testers

what if our ad hoc is the optimizing?



6 developers

2 testers

6 developers

~~32~~ testers

NTM - NOTAM Management

IFS Briefer - Globe

FOS - *Flight Operation Suite*



BRF - Briefer



ULD - ULD Control

WXS - Weather Station

QUE - *Communication & Alerting*

FIDS - Flight Information Display System



LST - Weight & Balance

FIDS - Flight Information Display System

CDM - Crew DX

FLW - Flight Watch

TOP - Aircraft Performance Tool



Microsoft®

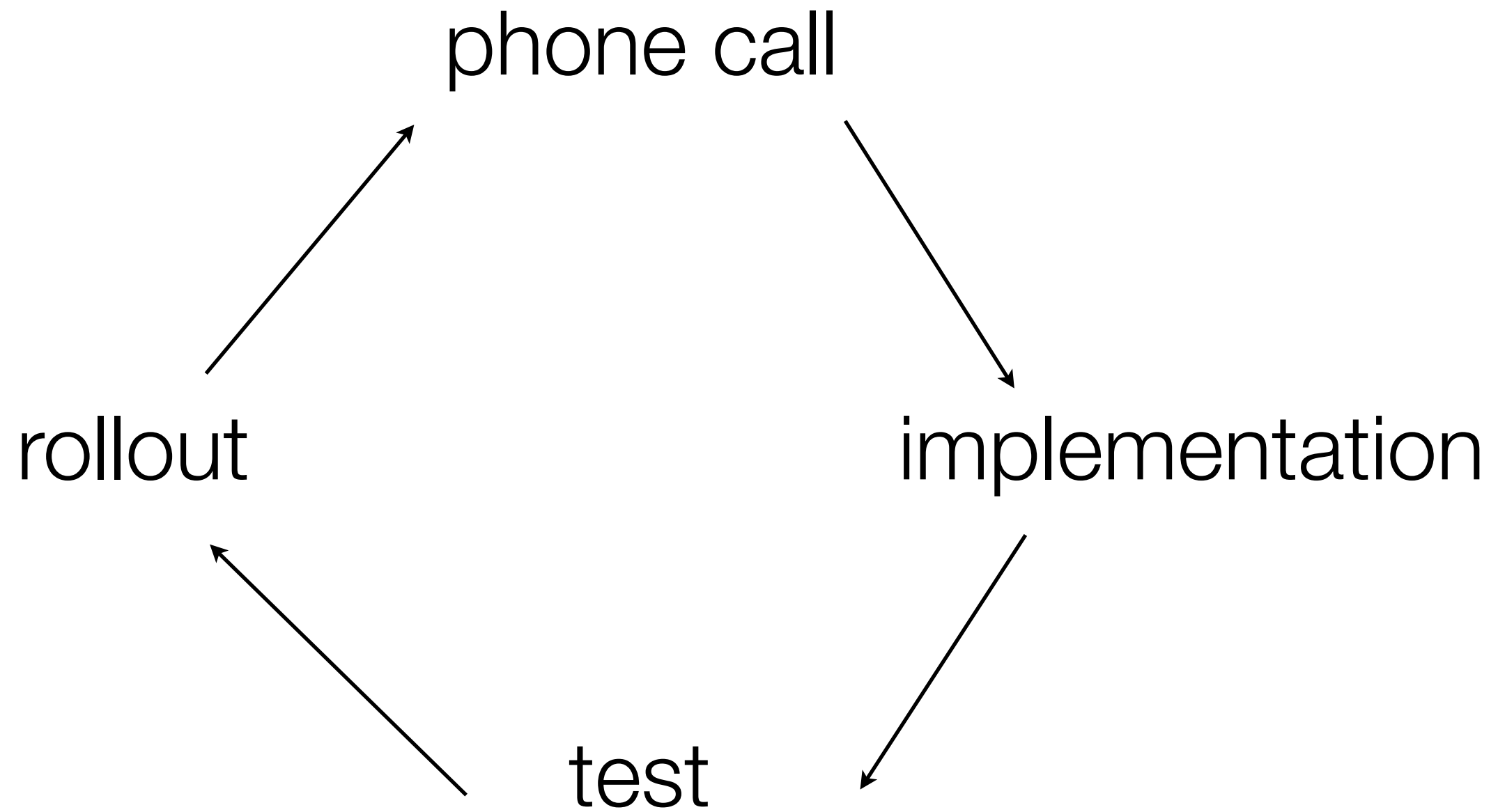
Office

Specialist



Specialist

no I didn't



Use Case: Receive Vehicle from Customer
Scope: System
Primary actor: Garage Representative (GR)

Basic Flow

1. GR informs system of vehicle being returned.
2. System presents vehicle rental information for verification by GR.
3. GR informs system that no vehicle issues (damage, mechanical problems) are observed, or reported by customer.
4. GR informs system of gasoline level in vehicle. System calculates re-fueling charge based on current level and any pre-payment agreement.
5. GR informs system to apply rental charge to credit card used on vehicle reservation.
6. System prints itemized receipt of rental charges.

Alternate Flows

2a. Not our vehicle

- 2a.1 GR informs customer this vehicle is not property of this rental company. This use case ends.

3a. Vehicle issues found

- 3a.1 GR informs system of problems reported by customer, or that damage is present.
- 3a.2 When damage is found, GR directs customer to branch Customer Service Representative to obtain details on damage.

5a. Apply charges to different credit card

- 5a.1 GR directs customer to branch Customer Service Representative to validate different credit card and transfer rental charges to that card.

ah, good times...

Agile



I learned it the hard way



Agile



Patterns are evil

A clerk at a butcher shop stands five feet ten inches tall and wears size 13 sneakers. What does he weigh?

In British Columbia you cannot take a picture of a man with a wooden leg. Why not?

Which is correct to say, “The yolk of the egg is white” or “The yolk of the egg are white?”

A clerk at a butcher shop stands five feet ten inches tall and wears size 13 sneakers. What does he weigh?

Meat

In British Columbia you cannot take a picture of a man with a wooden leg. Why not?

You can't take a picture with a wooden leg.

You need a camera (or iPad or cell phone) to take a picture.

Which is correct to say, “The yolk of the egg is white” or “The yolk of the egg are white?”

Neither. Egg yolks are yellow.



We see what we want to see and what we expect to see

Technology Radar



Techniques

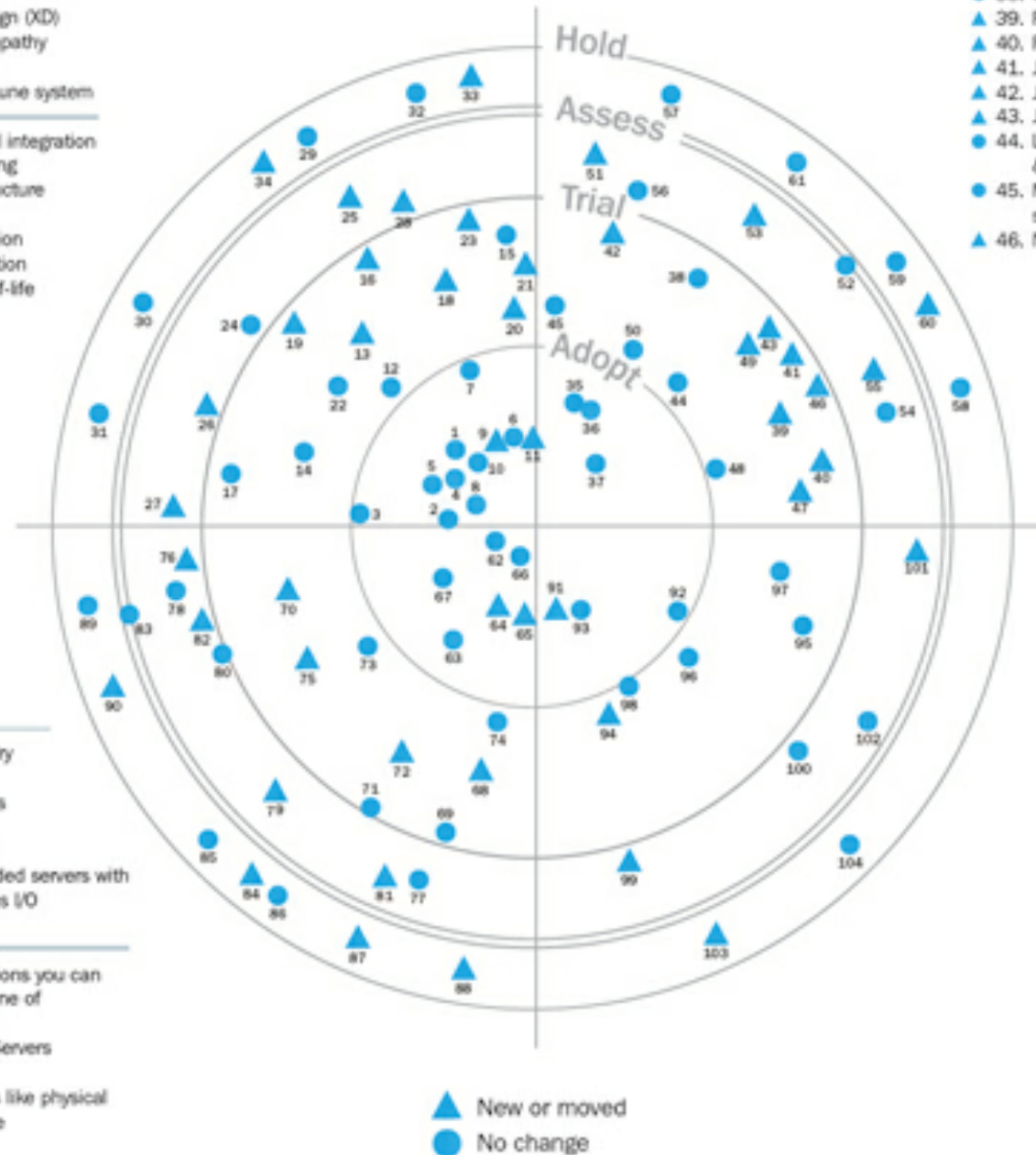
- Adopt**
 - 1. Automate database deployment
 - 2. Coding architects
 - 3. Continuous Delivery (CD)
 - 4. Data visualizations of development and operations
 - 5. DevOps
 - 6. Emergent design
 - 7. Evolutionary architecture
 - 8. Evolutionary database
 - 9. Health check pages
 - 10. Simple performance trending
 - 11. Test at the appropriate level
- Trial**
 - 12. Acceptance test of journeys
 - 13. Agile analytics
 - 14. Build your own radar
 - 15. Categorization & prioritization of technical debt
 - 16. Embedding a servlet container
 - 17. Event APIs
 - 18. Infrastructure automation of development workstations
 - 19. Out-of-container functional testing
 - 20. Performance testing as a first-class citizen
- Assess**
 - 21. Single command deploy
 - 22. Thoughtful caching
 - 23. Windows infrastructure automation
 - 24. Event sourcing
 - 25. Experience Design (XD)
 - 26. Mechanical sympathy
 - 27. Micro-services
 - 28. Production immune system
 - 29. Database based integration
 - 30. Feature branching
 - 31. Manual infrastructure management
 - 32. Scrum certification
 - 33. Server / application container end-of-life
 - 34. Test recorders
- Hold**
 - 35. Git
 - 36. Github
 - 37. Infrastructure as code

Platforms

- Adopt**
 - 62. ATOM
 - 63. AWS
 - 64. Care about hardware
 - 65. Communication between those responsible for hardware and software
 - 66. KVM
 - 67. Mobile web
- Trial**
 - 68. Domain-specific PaaS
 - 69. Heroku
 - 70. Linux containers
 - 71. Offline mobile webapps (just HTML5)
 - 72. Private clouds
 - 73. Tablet
 - 74. Ubiquitous computing
 - 75. Windows Phone 7
 - 76. AppHarbor
- Assess**
 - 77. Cloud Foundry
 - 78. GPGPU
 - 79. Hybrid clouds
 - 80. Node.js
 - 81. OpenSocial
 - 82. Single threaded servers with asynchronous I/O
 - 83. vFabric
 - 84. Buying solutions you can only afford one of
 - 85. GWT
 - 86. Java Portal Servers
 - 87. RIA
 - 88. Treating VMs like physical infrastructure
 - 89. WS-*
 - 90. Zero-code packages
- Hold**
 - 91. Care about languages
 - 92. HTML5
 - 93. JavaScript as a first-class language

Tools

- Adopt**
 - 35. Git
 - 36. Github
 - 37. Infrastructure as code
 - 38. Client-side MVC
 - 39. FPM
 - 40. Frank
 - 41. Jade
 - 42. JavaScript micro frameworks
 - 43. JavaScript tooling
 - 44. Log aggregation & indexing
 - 45. Message buses without smarts
 - 46. NuGet
- Trial**
 - 47. Polyglot persistence
 - 48. PowerShell
 - 49. PSake
 - 50. Vagrant
 - 51. Gradle
 - 52. jQuery Mobile
 - 53. Logic-free markup
 - 54. Open source BI/ETL tools
 - 55. Riak
 - 56. Sonar
- Assess**
 - 57. Code in configuration
 - 58. Cross-platform mobile toolkits
 - 59. Enterprise service bus
 - 60. Maven
 - 61. VCS with implicit workflow
- Hold**
 - 62. ATOM
 - 63. AWS
 - 64. Care about hardware
 - 65. Communication between those responsible for hardware and software
 - 66. KVM
 - 67. Mobile web
 - 68. Domain-specific PaaS
 - 69. Heroku
 - 70. Linux containers
 - 71. Offline mobile webapps (just HTML5)
 - 72. Private clouds
 - 73. Tablet
 - 74. Ubiquitous computing
 - 75. Windows Phone 7
 - 76. AppHarbor



Languages

- Adopt**
 - 91. Care about languages
 - 92. HTML5
 - 93. JavaScript as a first-class language
- Trial**
 - 94. Clojure
 - 95. CoffeeScript
 - 96. Domain-specific languages
 - 97. SASS, SCSS, and LESS
 - 98. Scala
- Assess**
 - 99. ClojureScript
 - 100. F#
 - 101. Functional Java
 - 102. Future of Java
- Hold**
 - 103. Google Dart
 - 104. Logic in stored procedures

I love the familiar. I categorize.

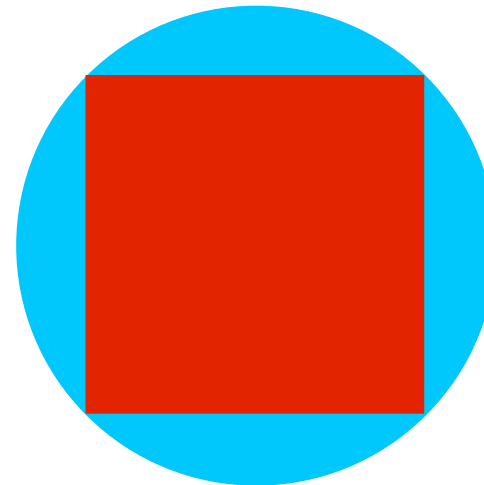
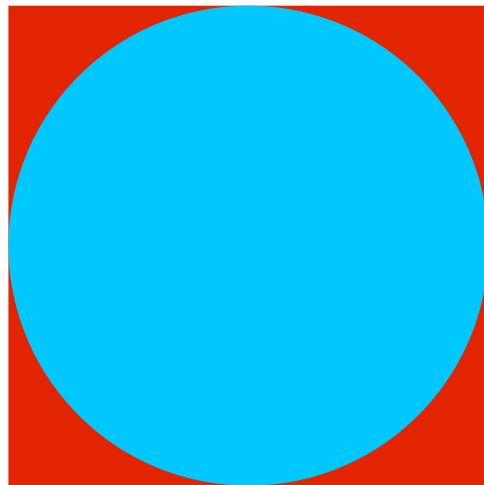
implement

observe

implementation

observe

Which fits better?



define "fits better"?

Emergent Design

quick solution

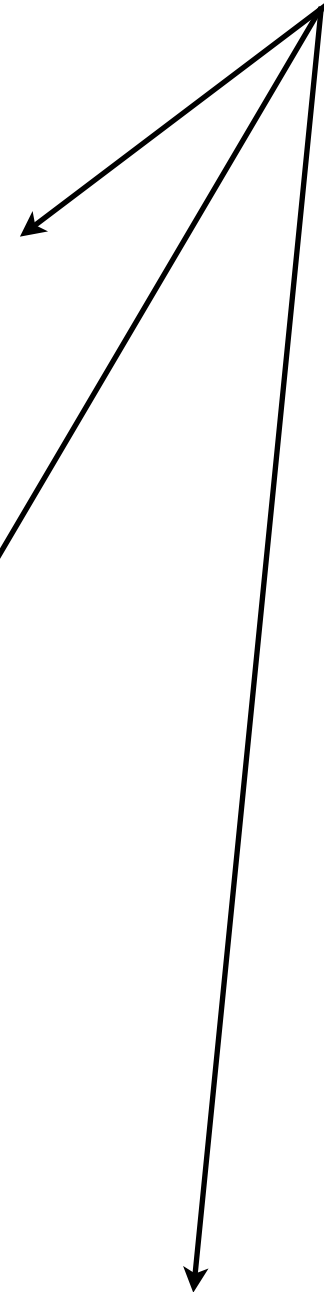
focus on the problem

code to compare

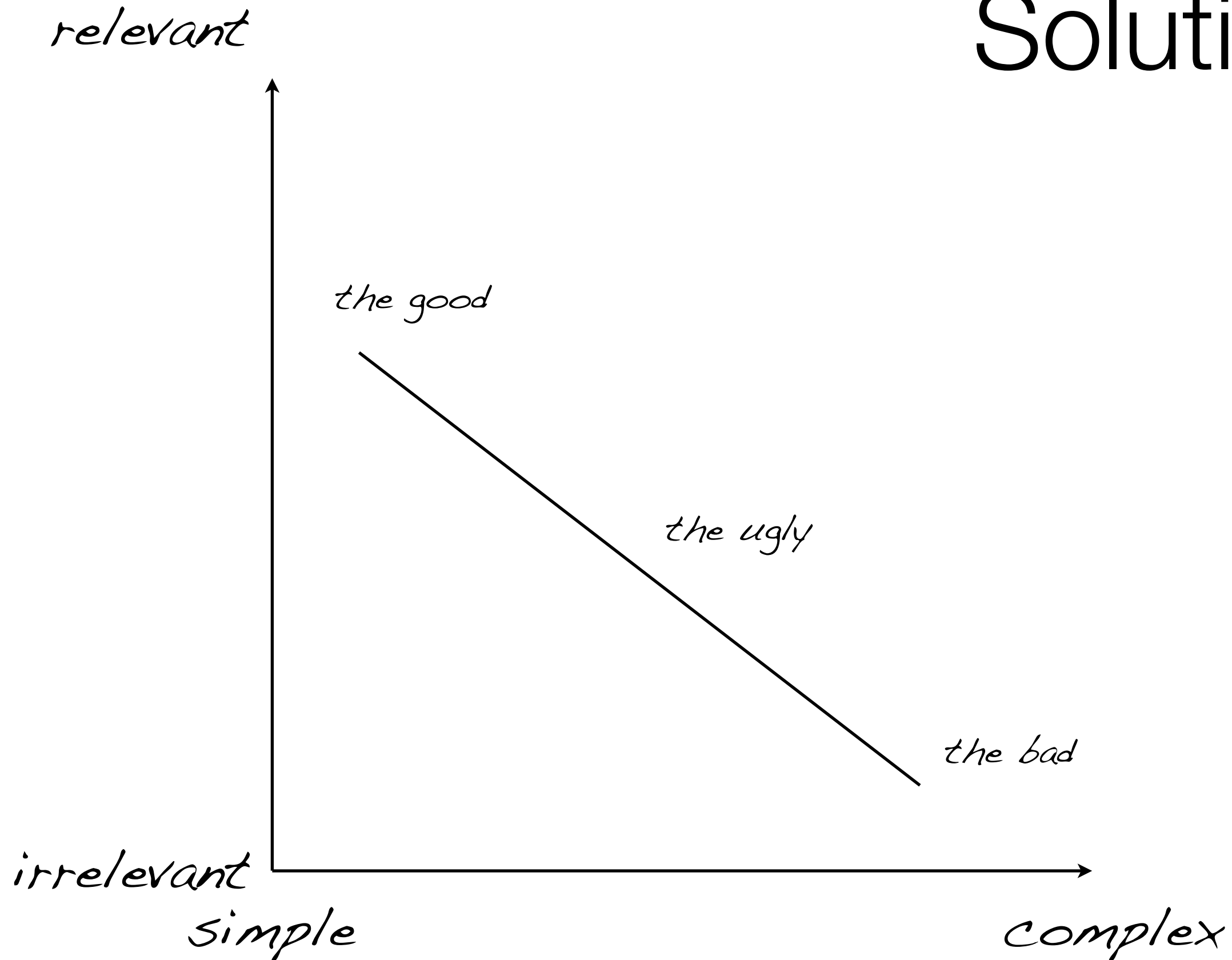
emerging patterns

things done better

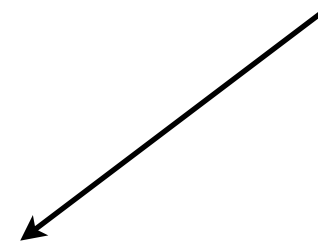
more than a sum of its parts



Solution

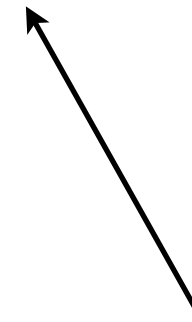


*focus on the what
(relevancy)*



1. Satisfy your immediate needs.

2. Observe the patterns and adapt the solution.



*focus on the
how (simplicity)*

```
public void setProduct(String productAlias) {  
    this.product = Product.from(productAlias);  
}
```

relevant? simple?


```
public void setProduct(Product product) {  
    this.product = product;  
}
```

```
public class ProductEnumConverter extends PropertyEditorSupport {  
    @Override  
    public void setAsText(final String productAlias) throws IllegalArgumentException {  
        setValue(Product.from(productAlias));  
    }  
}
```

```
@InitBinder  
public void initBinder(WebDataBinder binder) {  
    binder.registerCustomEditor(Product.class, new ProductEnumConverter());  
}
```

```
<bean class="org.springframework.beans.factory.config.CustomEditorConfigurer">  
    <property name="customEditors">  
        <map>  
            <entry key="domain.model.product.Product" value="domain.infrastructure.Produc<br/>  
        </map>  
    </property>  
</bean>
```

relevant? simple?

+

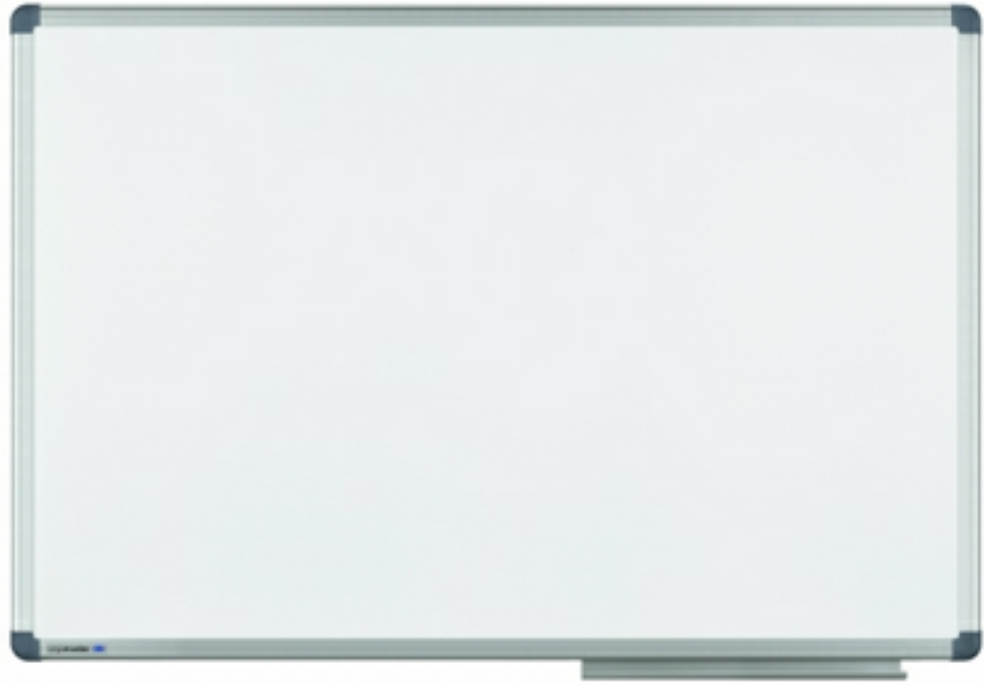


Agile

Agile



Agile

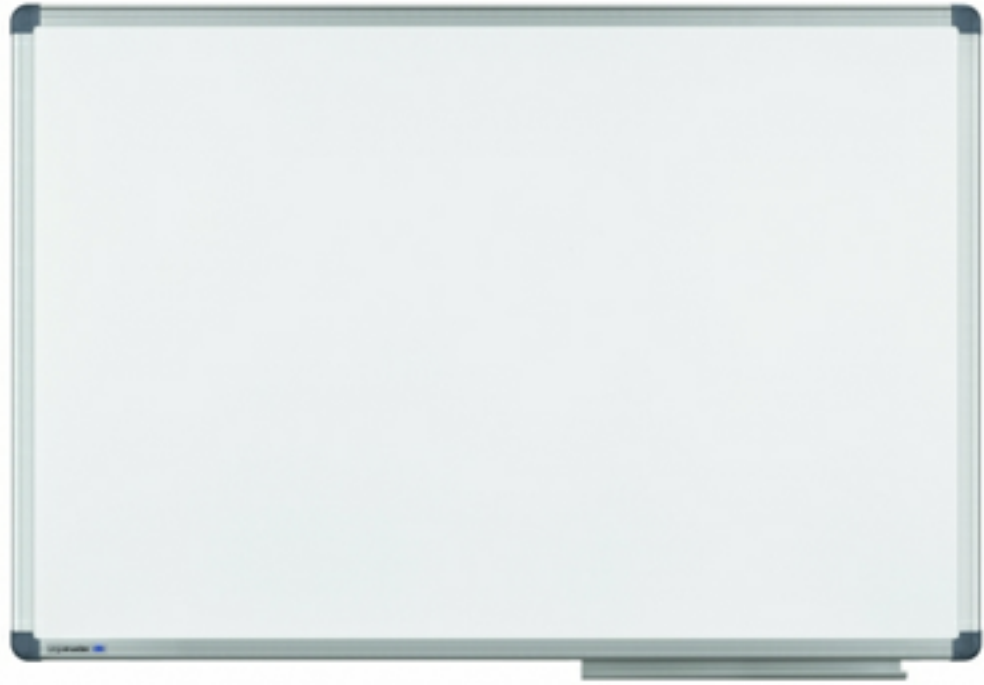


Agile



Agile





Agile



is it? Scrum much?

Question 1 - Iterations

No iterations - 0

Iterations > 6 weeks - 1

Variable length < 6 weeks - 2

Fixed iteration length 6 weeks - 3

Fixed iteration length 5 weeks - 4

Fixed iteration 4 weeks or less - 10

Question 3 - Agile Specification

No requirements - 0

Big requirements documents - 1

Poor user stories - 4

Good requirements - 5

Good user stories - 7

Just enough, just in time specifications - 8

Good user stories tied to specifications as needed - 10

You loose 2 points if your requirements are good but not user stories

Question 6 - Estimates

Product Backlog not estimated - 0

Estimates not produced by team - 1

Estimates not produced by planning poker - 5

Estimates produced by planning poker by team -
8

Estimate error $< 10\%$ - 10

“We give this test to Scrum teams at OpenView Venture Partners and to their portfolio companies as the venture group **does not expect good performance from Scrum teams without passing the Nokia test.** They are also very interested in **predictability** of release dates which is **impossible without passing grades on the test.**”

Jeff Sutherland

“The Nokia Test? I hope you’re not expecting to graduate just because you passed.”

*Brad Murphy,
Founder & CEO of Gear Stream*

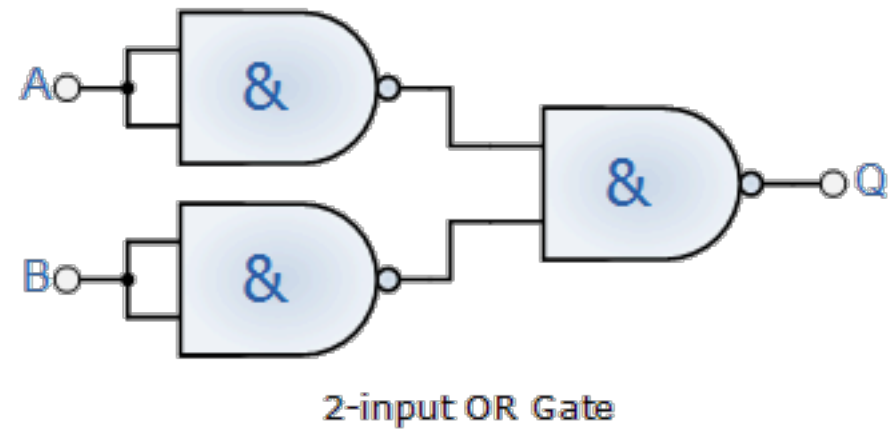
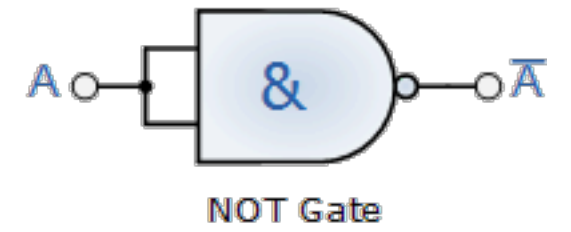
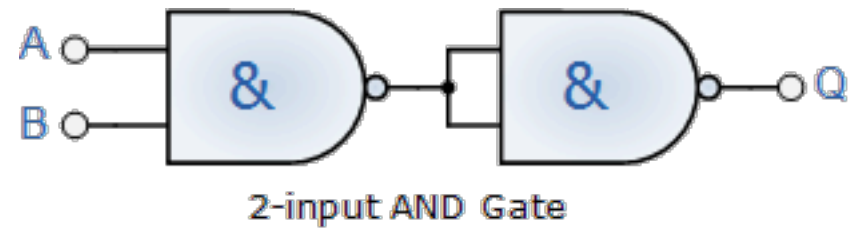
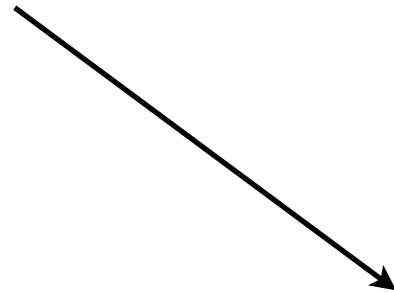


Matryoshka syndrome





We can't change it, because then it's not Scrum anymore.

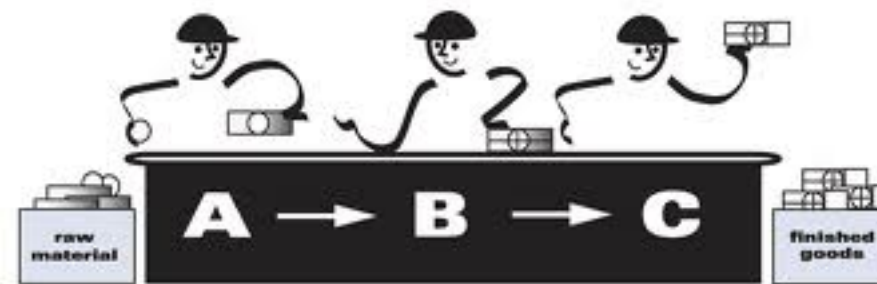


Functional completeness

AGILE

Manifesto for Agile Software Dev.

- INDIVIDUALS AND INTERACTIONS OVER PROCESSES AND TOOLS
- WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTATION
- CUSTOMER COLLABORATION OVER CONTRACT NEGOTIATION
- RESPONDING TO CHANGE OVER FOLLOWING A PLAN



Functional completeness

Agile

=

Common sense

Open space

Great People

Trust

Pair programming

In-house coaching

Experimenting

Individuals & interactions

(Passionate) discussions

Flat structure

Outings

Visibility

And a pinch of crazy



The team

Continuous deployment

Hive mind

Monitoring

Emergent design

Experimenting

Working software

TDD

Evolutionary architecture

DDD

Visibility

BDD

Trust

git push

Uitgelicht

- 
Klussen
Plaats uw klus
- 
Auto's
150.000+ occasions
- 
Aanbiedingen
Tot wel 85% korting
- 
Introductie
Testversie

Groepen

[Antiek en Kunst](#)
[Audio, Tv en Foto](#)
[Auto's](#)
[Auto-onderdelen](#)
[Auto diversen](#)
[Boeken](#)
[Caravans en Kamperen](#)
[Cd's en Dvd's](#)
[Computers en Software](#)
[Contacten en Berichten](#)
[Diensten en Vakmensen](#)
[Dieren en Toebehoren](#)



Auto's



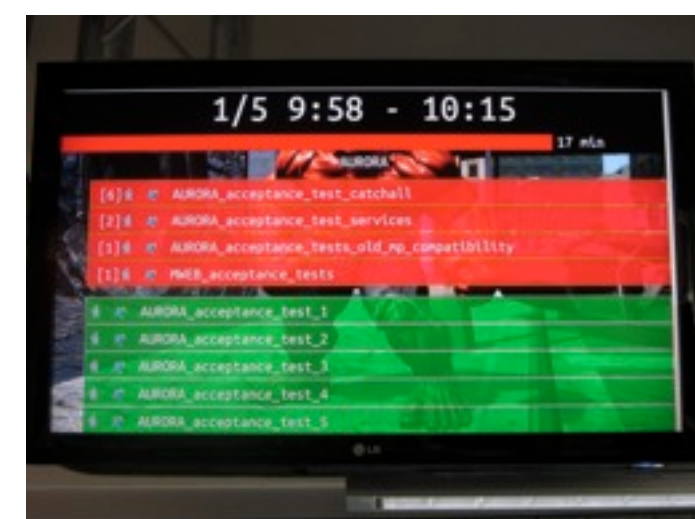
Peugeot Partner 170C 1.9
 € 1.599,00
 Zojuist geplaatst: Vandaag 20:32



Volkswagen Polo 1.4 74K...
 € 5.250,00
 Zojuist geplaatst: Vandaag 20:31



Peugeot 807 2.0
 € 3.750,00
 Zojuist geplaatst: V



One team

User testing

Beta

Flat structure

Experimenting

Customer collaboration

Outings

Open space

Hive mind

Visibility

Trust

(Passionate) discussions

One for all - all for one



Last responsible moment

TDD

Continuous deployment

Experimenting

Responding to change

Hive mind

Let it hurt

Trust

Real priorities

Visibility

One for all - all for one



Be Agile.





bkiera@ebay.com



@useresu



mutablenotions.wordpress.com



Neal Ford

tinyurl.com/nf-ead

developerWorks®

<http://www.thoughtworks.com/emergent-design>

ThoughtWorks®

More?



Start complaining. That was your training.

Thank You!