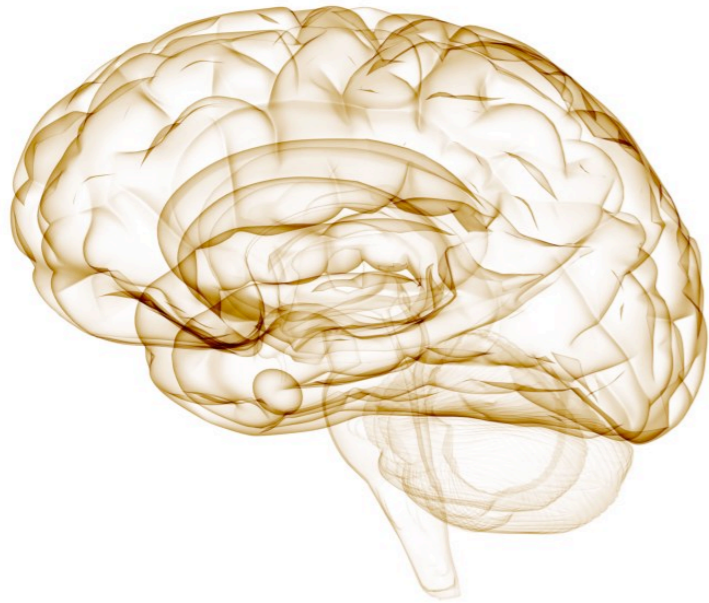


Behaviour-Driven Development

Writing software that matters

Aslak Helleøy - Chief Scientist - BEKK










Stakeholders



Programmers

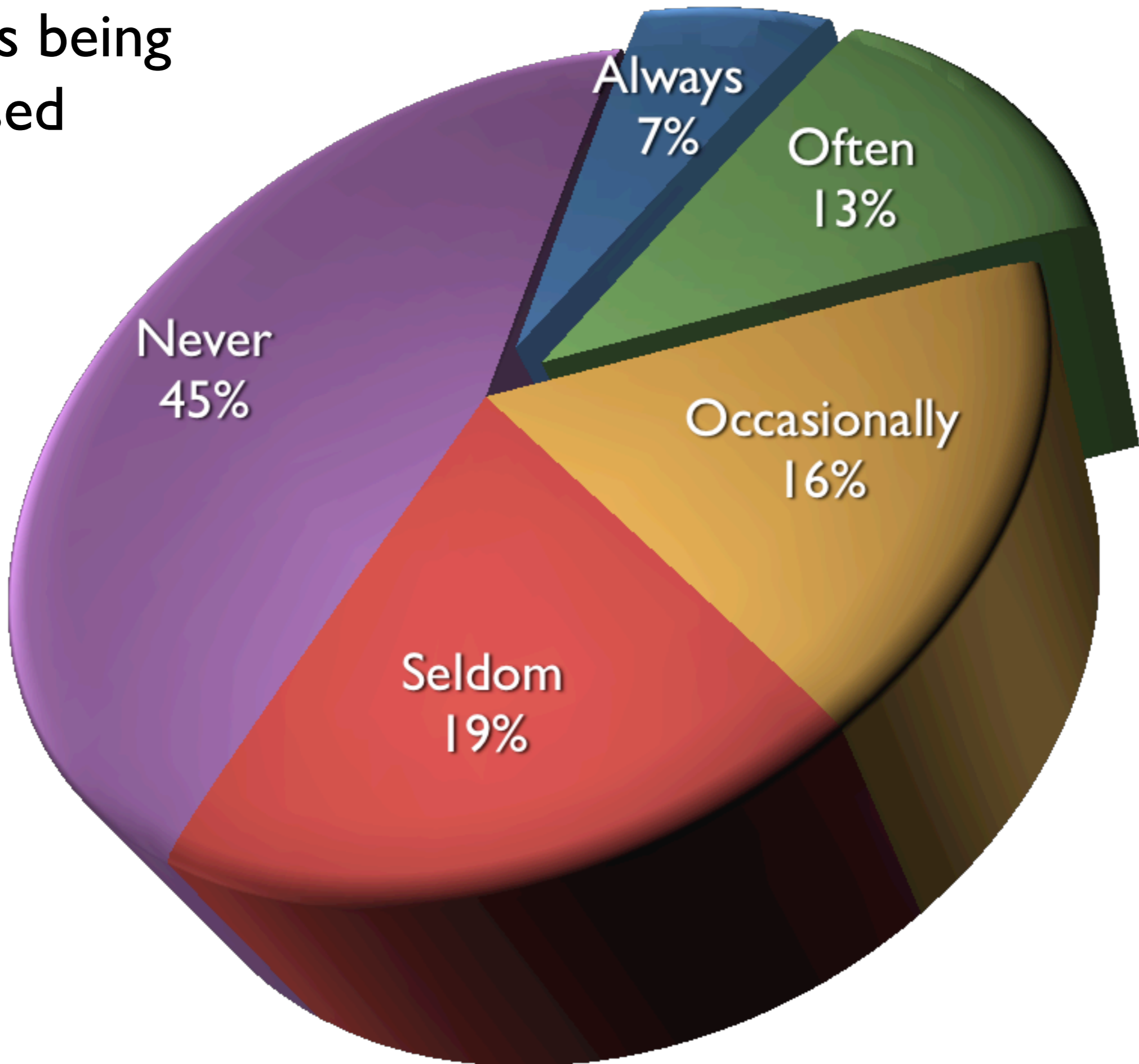
Dual audience

Origin and inspiration

 <p>JBehave Dan North Chris Matts</p>	 <p>Test-Driven Development Kent Beck</p>	 <p>FIT Ward Cunningham</p>
 <p>User Stories Mike Cohn</p>	 <p>Example Driven Development Brian Marick</p>	 <p>Chelimsky, Hellesøy, Takita, Astels, Baker</p>
 <p>Connextra Stories Several</p>	 <p>Domain-Driven Design Eric Evans</p>	 <p>Acceptance-Test Driven Planning</p>

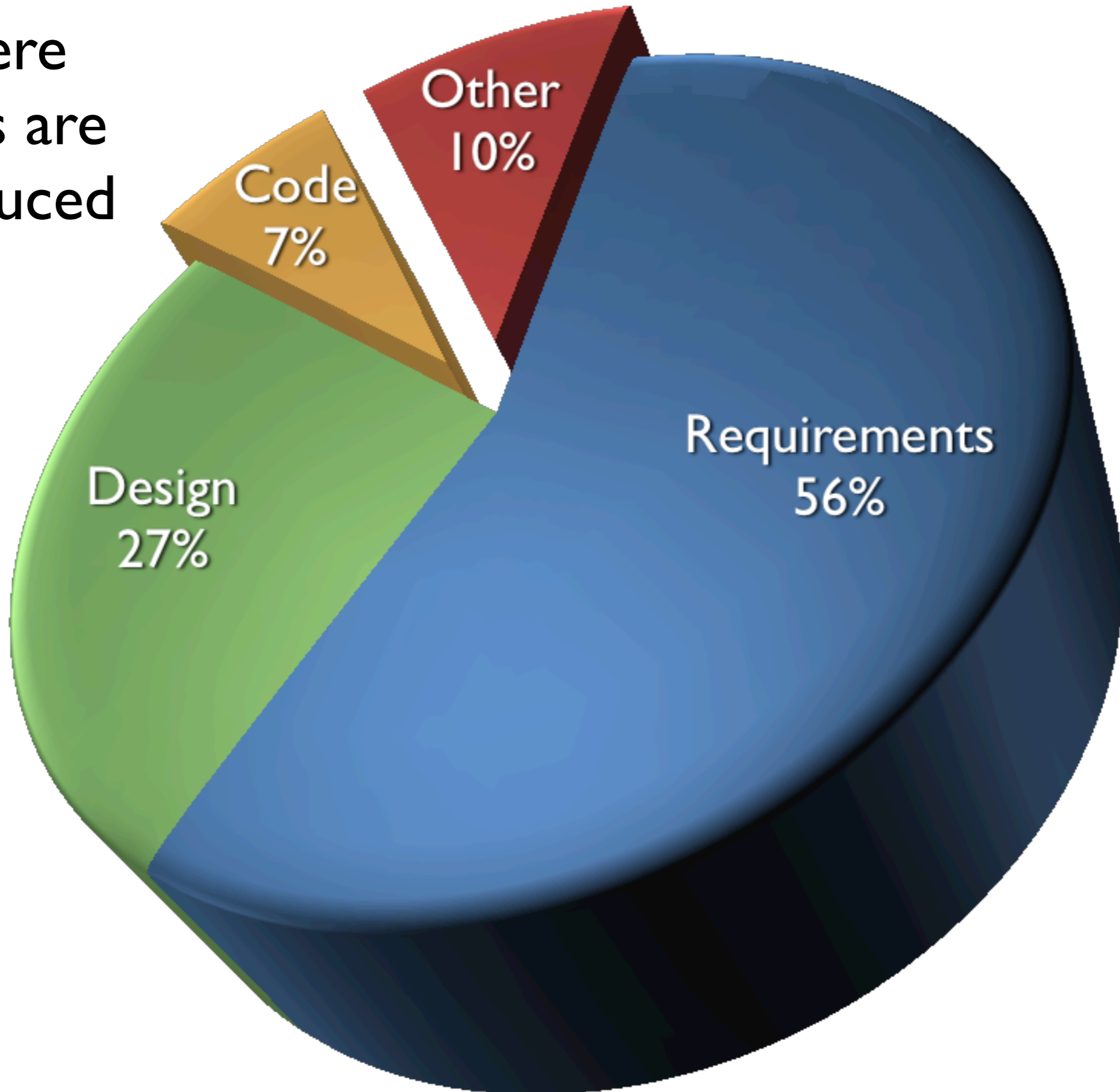
Software development is
all about delivering
business value

What's being used



Source: CHAOS report

Where
errors are
introduced



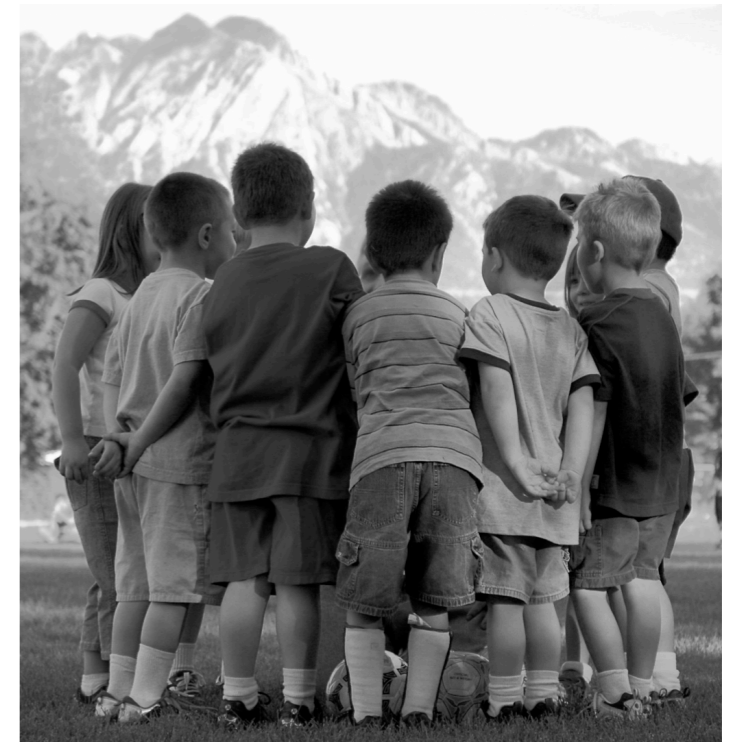
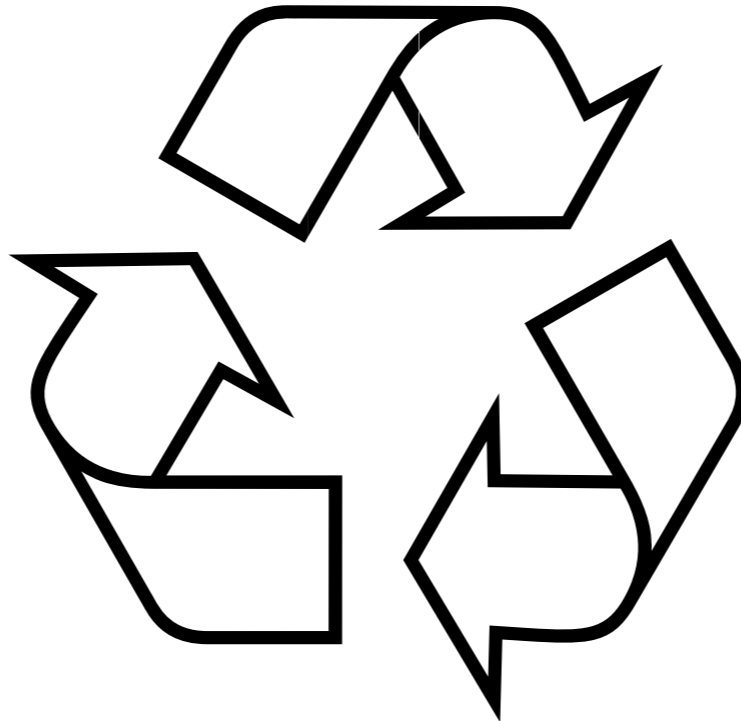
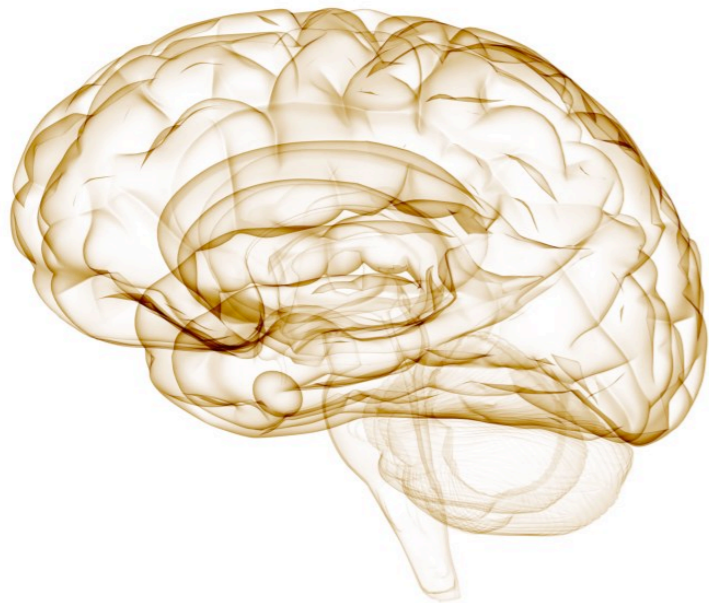
Source: CHAOS report



Trumpet Call
Made by Hohner

MADE BY H. HOHNER GERMANY

BDD: Business value



Ubiquitous Language

BDD User Stories

Withdraw Cash

Role

As an **Account Holder**

Operation



I want to **withdraw cash in an ATM**

So that **I can get cash when the bank
is closed**

1

Business value

Origin: Connextra

Perspective	Title	Reserved for priority
	WRITING GOOD STORIES	
Reason	As a Connextra employee - I want to know how to write good stories so that I can submit cards to the planning game that are clear and will be accepted in the next iteration.	
		Requirements
Author	Date	Reserved for estimate
Tim	8/Nov/01	

Challenge:
Common understanding
of “Done”

BDD Scenarios

Given I have £200 in my account

When I ask to withdraw £20

Then I should be given £20

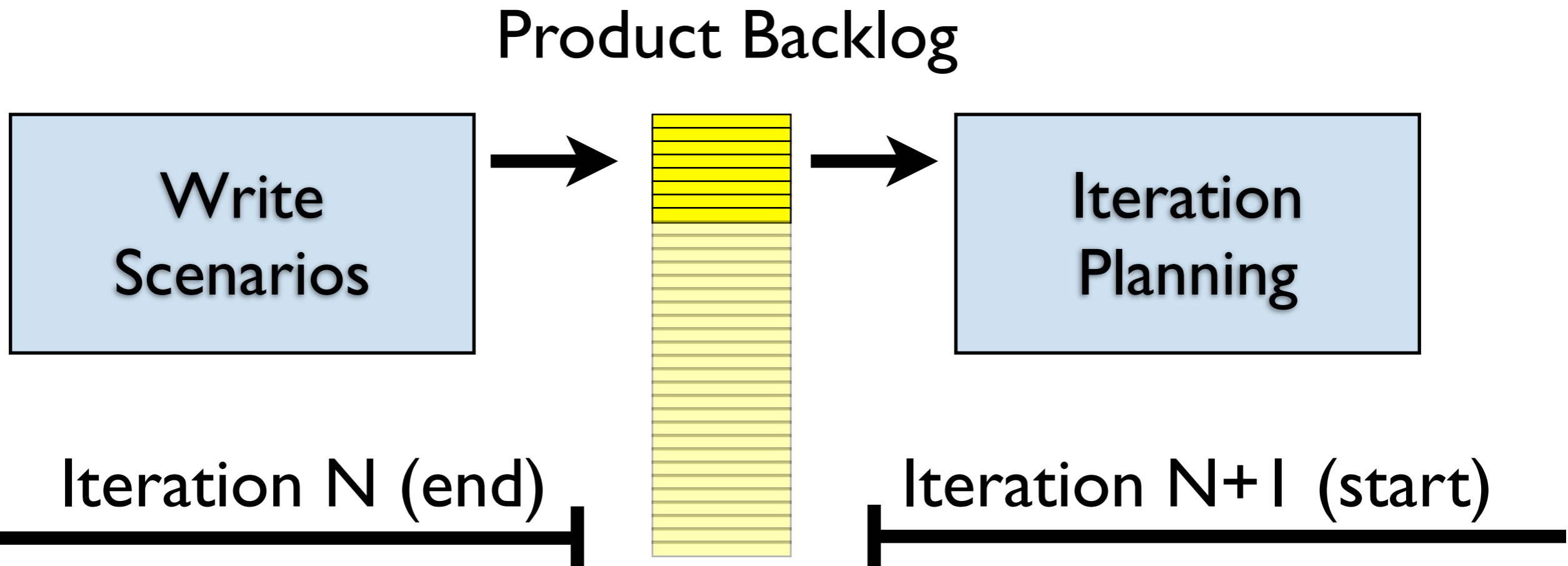
And my balance should be £180

1

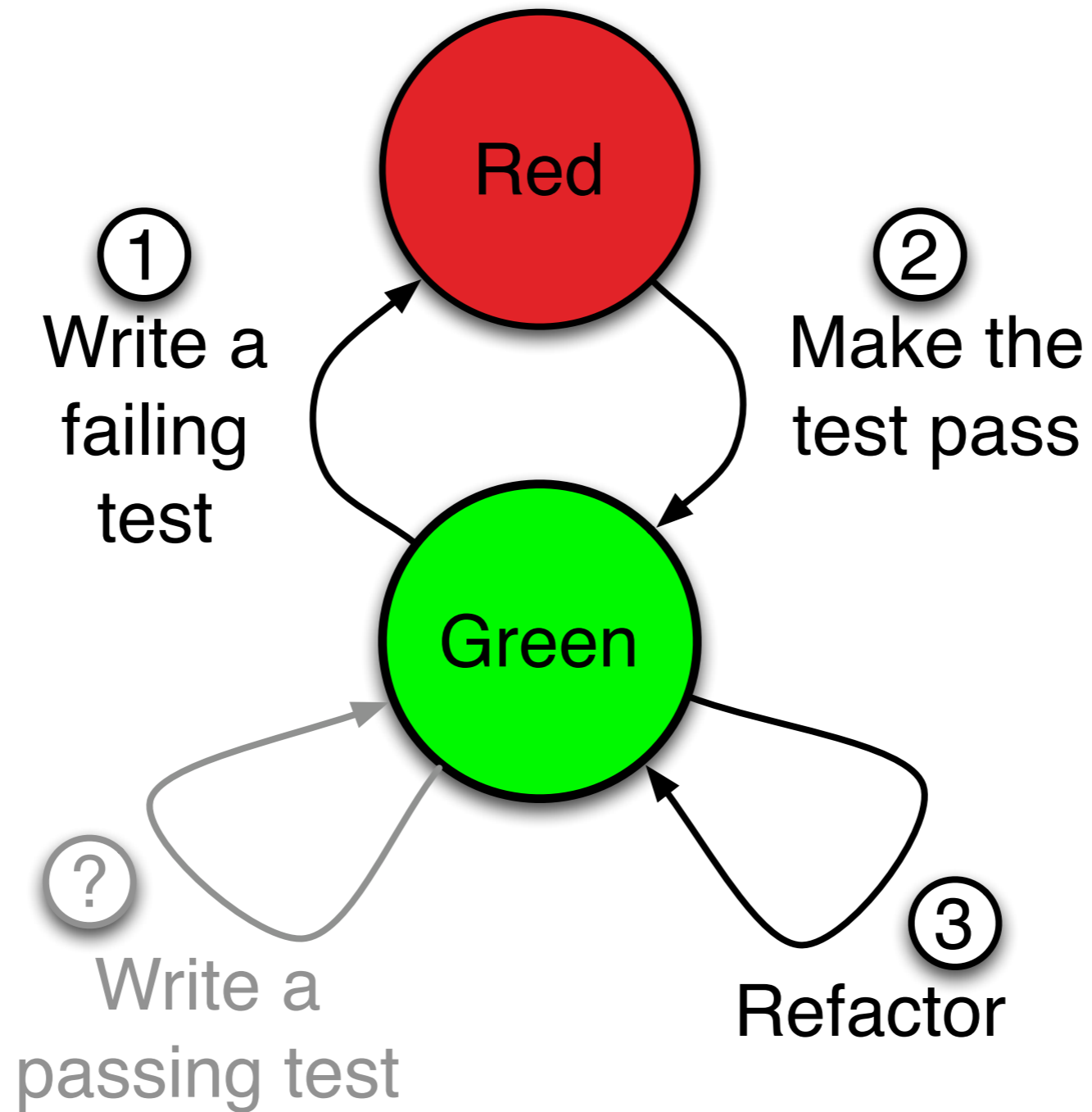
Business value

ATDP

Acceptance-Test Driven Planning



TDD



Developers don't know...

- Where to start
 - What to test
 - What not to test
 - How much to test in one go
 - What to call tests
 - How to fix a broken test
 - That TDD is about **design**
- 

Outside-In

Perspective	Title	Reserved for priority
	WRITING GOOD STORIES	
Reason	As a Connextra employee - I want to know how to write good stories so that I can submit cards to the planning game that are clear and will be accepted in the next iteration.	Requirements
Author	Tom	8/Nov/01
		Reserved for estimate

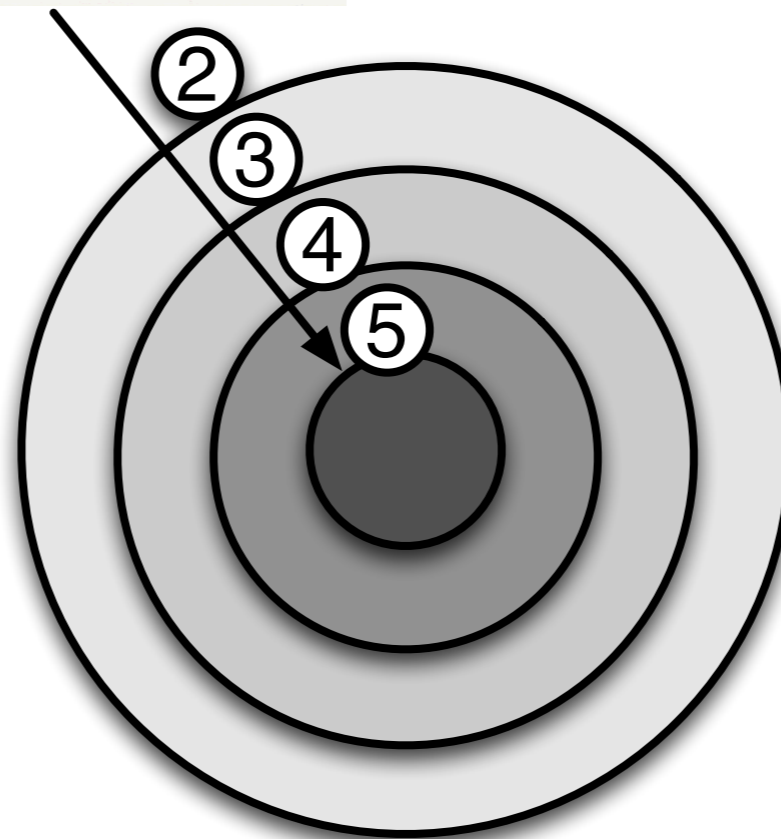
1

2

3

4

5

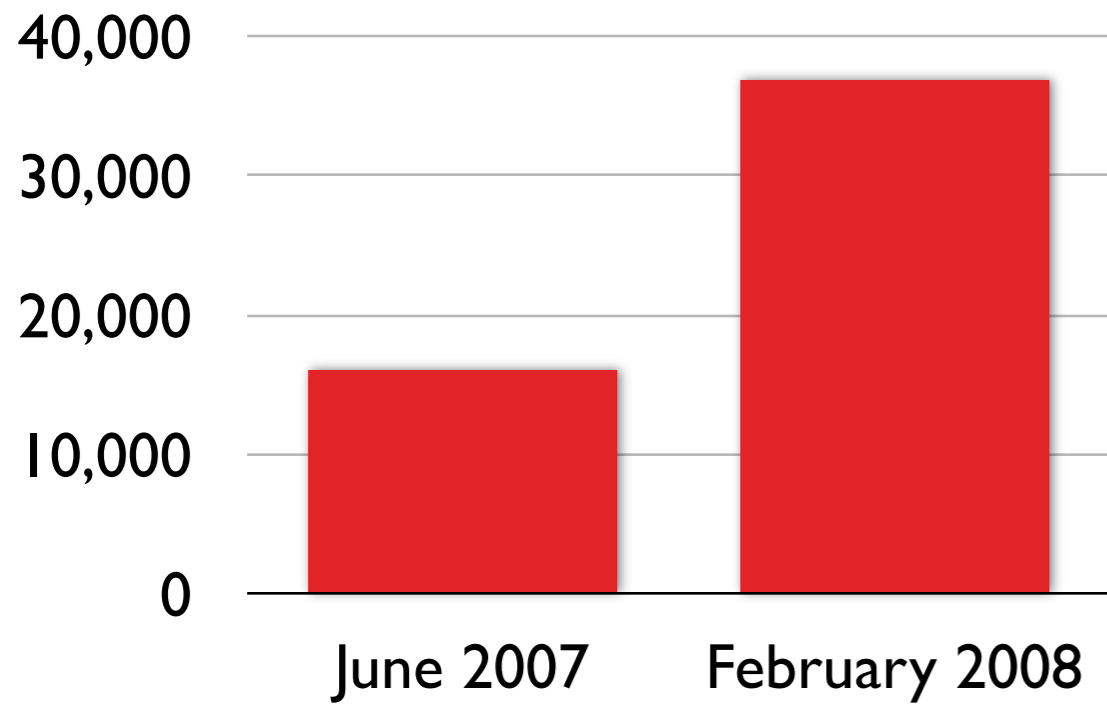


describe it with

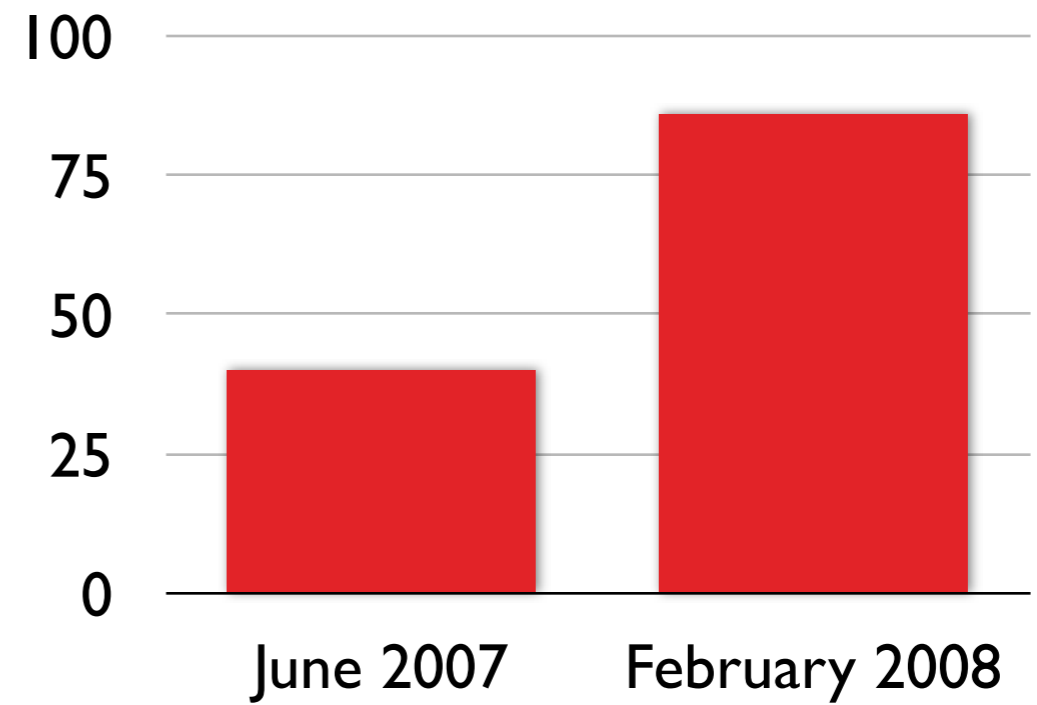
rspec



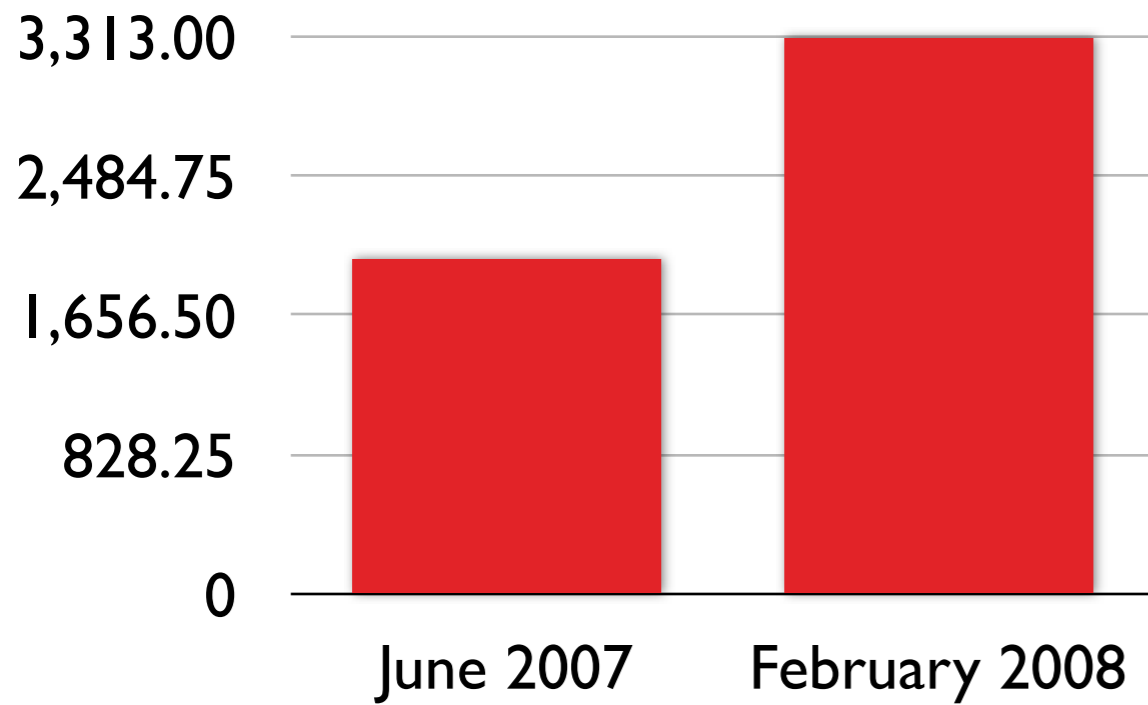
Downloads



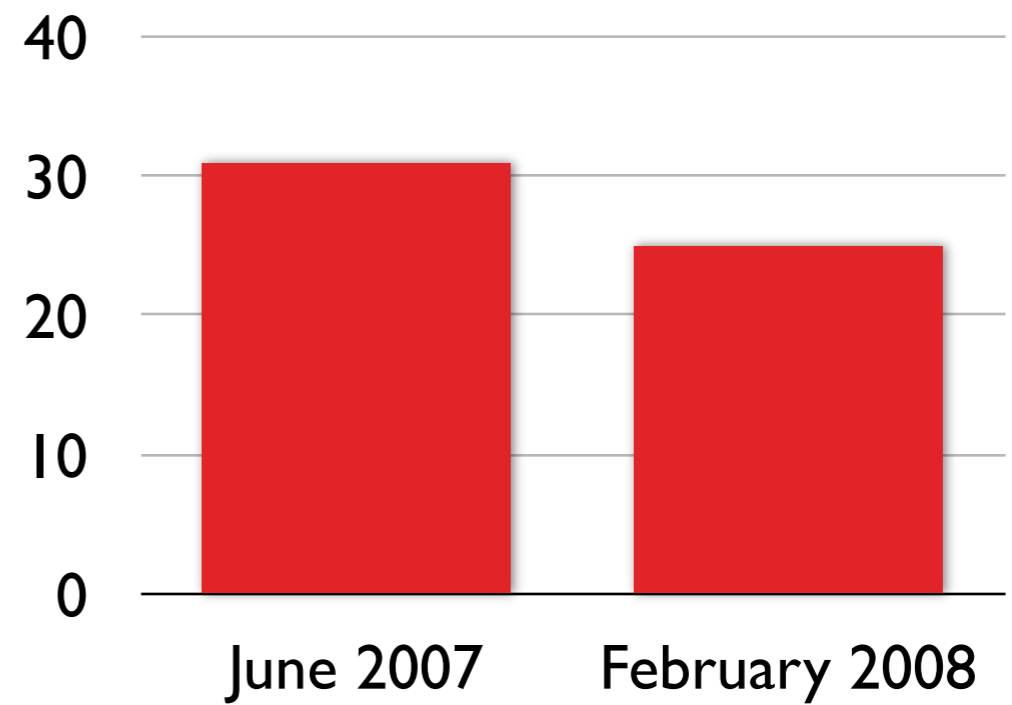
Contributors



SVN commits



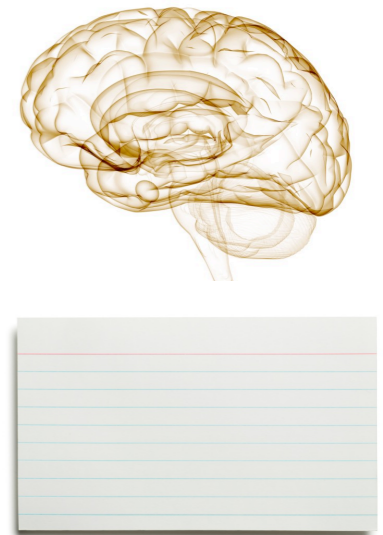
RubyForge rank



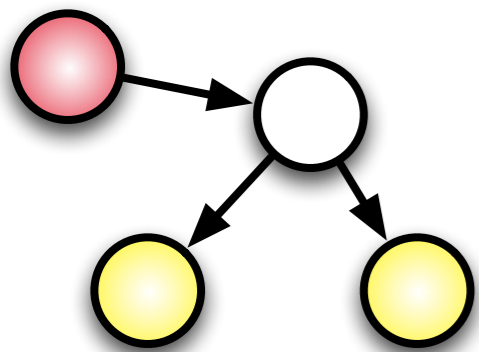
JRuby << RSpec

RSpec consists of 2 parts

Story Framework



Example Framework



RSpec *User Stories*

Story: Log in

As a Project contributor

I want to log in

So that I can see what projects I have access to

Scenario: User provides wrong credentials

Given I am on the login page

When I log in with login nobody and password useless

Then I should be informed that my password is incorrect

And I should not see aslak.hellesoy on the page

Scenario: User provides correct credentials

Given I am on the login page

When I log in with login aslak.hellesoy and password temporary

Then I should see aslak.hellesoy on the page

And I should have the opportunity to log out

RSpec *Examples*

```
require 'bowling'

describe Bowling do
  before(:each) do
    @bowling = Bowling.new
  end

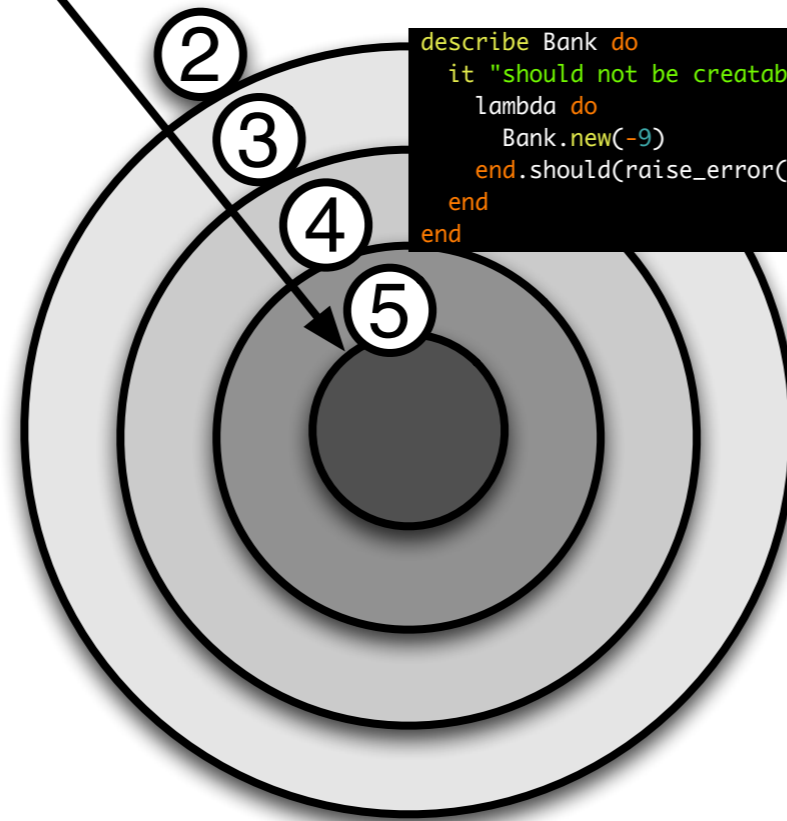
  it "should score 0 for gutter game" do
    20.times { @bowling.hit(0) }
    @bowling.score.should == 0
  end
end
```


Outside-In



```
Story: Withdraw Money ①  
As an account holder  
I want to withdraw money  
So that I can get cash when the bank is closed  
  
Scenario: Overdraft account  
Given an account with -1000 kr  
When I ask for 200 kr  
Given the balance should be -1000 kr
```

```
describe Bank do  
  it "should not be creatable with negative balance" do  
    lambda do  
      Bank.new(-9)  
    end.should(raise_error("Negative balance"))  
  end  
end
```



Summary

- As a <Role>, I want <Operation> So that <Business value>
- Given <Known state>, When I <Action>, Then <Expected outcome>
- Work Outside-in in the layer onion
- Write scenarios and tests bottom up (start with outcome)
- Tests are sentences

References

- <http://behaviour-driven.org/>
- <http://dannorth.net/introducing-bdd/>
- <http://dannorth.net/whats-in-a-story/>
- <http://rspec.info/>
- <http://jtestr.codehaus.org/>
- <http://jruby.codehaus.org/>
- <http://blog.aslakhellesoy.com/>