Staying Ahead of the Curve

Trisha Gee, Java Developer at MongoDB

or:
New Technology is
Dangerous, mmmkay

Let’s Learn Lambdas!

erm…

Let’s Go Parallel!

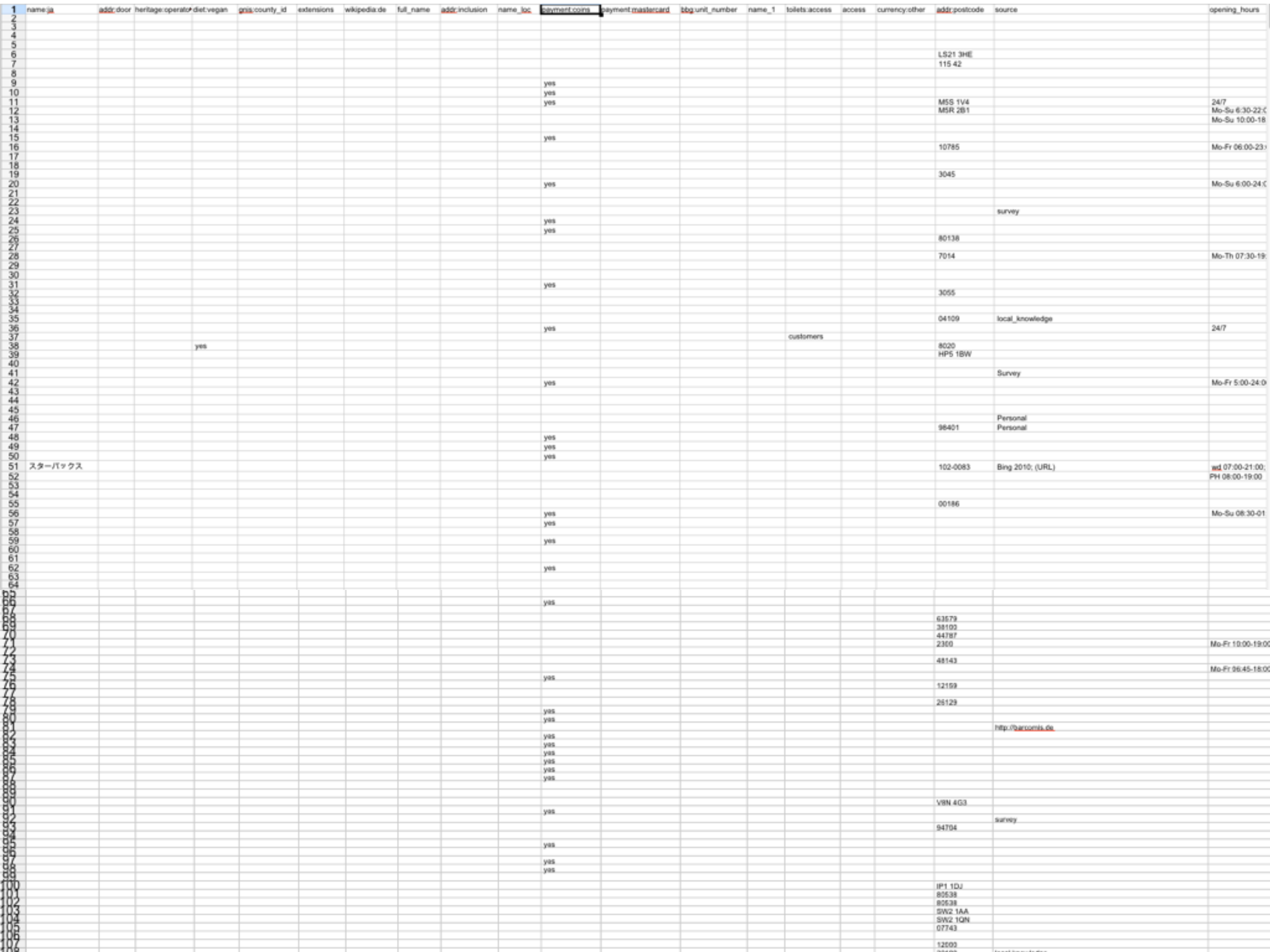
Wait a minute…

Why?

For The Business

Why?

- Customisable dashboards
- A new market squeezes your downtime window
- Trying to store different shaped data in a single table



1	name:ja	addr:door	heritage:operato	diet:vegan	gnis:county_id	extensions	wikipedia:de	full_name	addr:inclusion	name_loc	payment:coins	payment:mastercard	bbg:unit_number	name_1	toilets:access	access	currency:other	addr:postcode	source	opening_hours
2																				
3																				
4																				
5																				
6																		LS21 3HE		
7																		115 42		
8																				
9											yes									
10											yes									
11											yes							M5S 1V4		24/7
12																		M5R 2B1		Mo-Su 6:30-22:
13																				Mo-Su 10:00-18
14																				
15											yes									
16																		10785		Mo-Fr 06:00-23:
17																				
18																				
19																		3045		
20											yes									Mo-Su 6:00-24:
21																				
22																				
23																			survey	
24											yes									
25											yes									
26																		80138		
27																				
28																		7014		Mo-Th 07:30-19:
29																				
30																				
31											yes									
32																		3055		
33																				
34																				
35																		04109	local_knowledge	
36											yes									24/7
37															customers					
38				yes														8020		
39																		HP5 1BW		
40																				
41																			Survey	
42											yes									Mo-Fr 5:00-24:0
43																				
44																				
45																				
46																			Personal	
47																		98401	Personal	
48											yes									
49											yes									
50											yes									
51	スターバックス																	102-0083	Bing 2010; (URL)	wd 07:00-21:00;
52																				PH 08:00-19:00
53																				
54																				
55																		00186		
56											yes									Mo-Su 08:30-01
57											yes									
58																				
59											yes									
60																				
61																				
62											yes									
63																				
64																				
65																				
66											yes									
67																				
68																		63579		
69																		38103		
70																		44787		
71																		2300		Mo-Fr 10:00-19:00
72																				
73																		48143		
74																				Mo-Fr 06:45-18:00
75											yes									
76																		12159		
77																				
78																		26129		
79											yes									
80											yes									
81																			http://barcomis.de	
82											yes									
83											yes									
84											yes									
85											yes									
86											yes									
87											yes									
88																				
89																				
90																		V8N 4G3		
91											yes									
92																			survey	
93																		94704		
94																				
95											yes									
96																				
97											yes									
98											yes									
99																				
100																		IP1 1DJ		
101																		80538		
102																		80538		
103																		SW2 1AA		
104																		SW2 1QN		
105																		07743		
106																				
107																		12003		
108																		[illegible]	local_knowledge	

Now What?

- Identify possible technology options, e.g:
    - HTML5/JavaScript
    - DevOps tools
    - NoSQL

```
{
    "_id" : ObjectId("54536368300449768e8db864"),
    "openStreetMapId" : "20980098",
    "location" : {
        "coordinates" : [
            151.2728552,
            -33.8931988
        ],
        "type" : "Point"
    },
    "amenity" : "cafe",
    "cuisine" : "coffee_shop",
    "name" : "Gloria Jean's"
}
{
    "_id" : ObjectId("54536368300449768e8db866"),
    "openStreetMapId" : "21554712",
    "location" : {
        "coordinates" : [
            151.9519968,
            -27.5601496
        ],
        "type" : "Point"
    },
    "amenity" : "cafe",
    "created_by" : "JOSM",
    "cuisine" : "coffee_shop",
    "description" : "The only place in Toowoomba that roasts its own coffee",
    "name" : "Chipiron Cafe"
}
{
    "_id" : ObjectId("54536368300449768e8db867"),
    "openStreetMapId" : "27653001",
    "location" : {
        "coordinates" : [
            -1.6937433,
            53.9043868
        ],
        "type" : "Point"
    },
    "addr:city" : "Otley",
    "addr:country" : "GB",
    "addr:housenumber" : "9",
    "addr:postcode" : "LS21 3HE",
    "addr:street" : "Mercury Row",
    "amenity" : "cafe",
    "cuisine" : "coffee_shop",
    "gluten_free" : "yes",
    "is_in:town" : "Otley",
    "name" : "The Walkway Cafe"
}
{
    "_id" : ObjectId("54536368300449768e8db865"),
    "openStreetMapId" : "21321517",
```

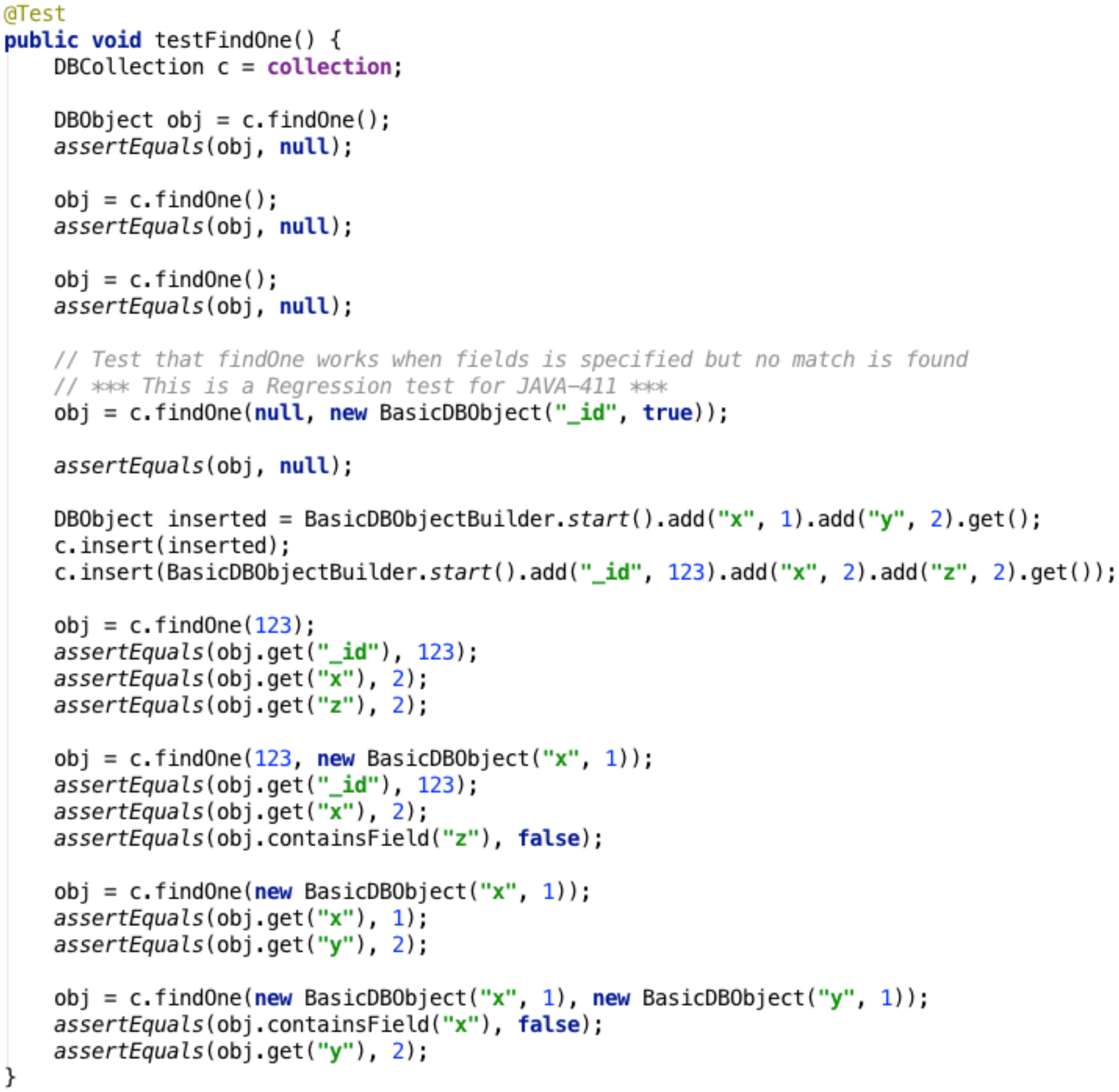
Approaches

- Spikes / prototypes
- Use in tests
- Use in non-production code
    - Build / Deploy / Continuous Delivery Pipeline
- Use in internal tools
- Use on a new project
- Get the experts in (remember to share)

For The Team

Why?

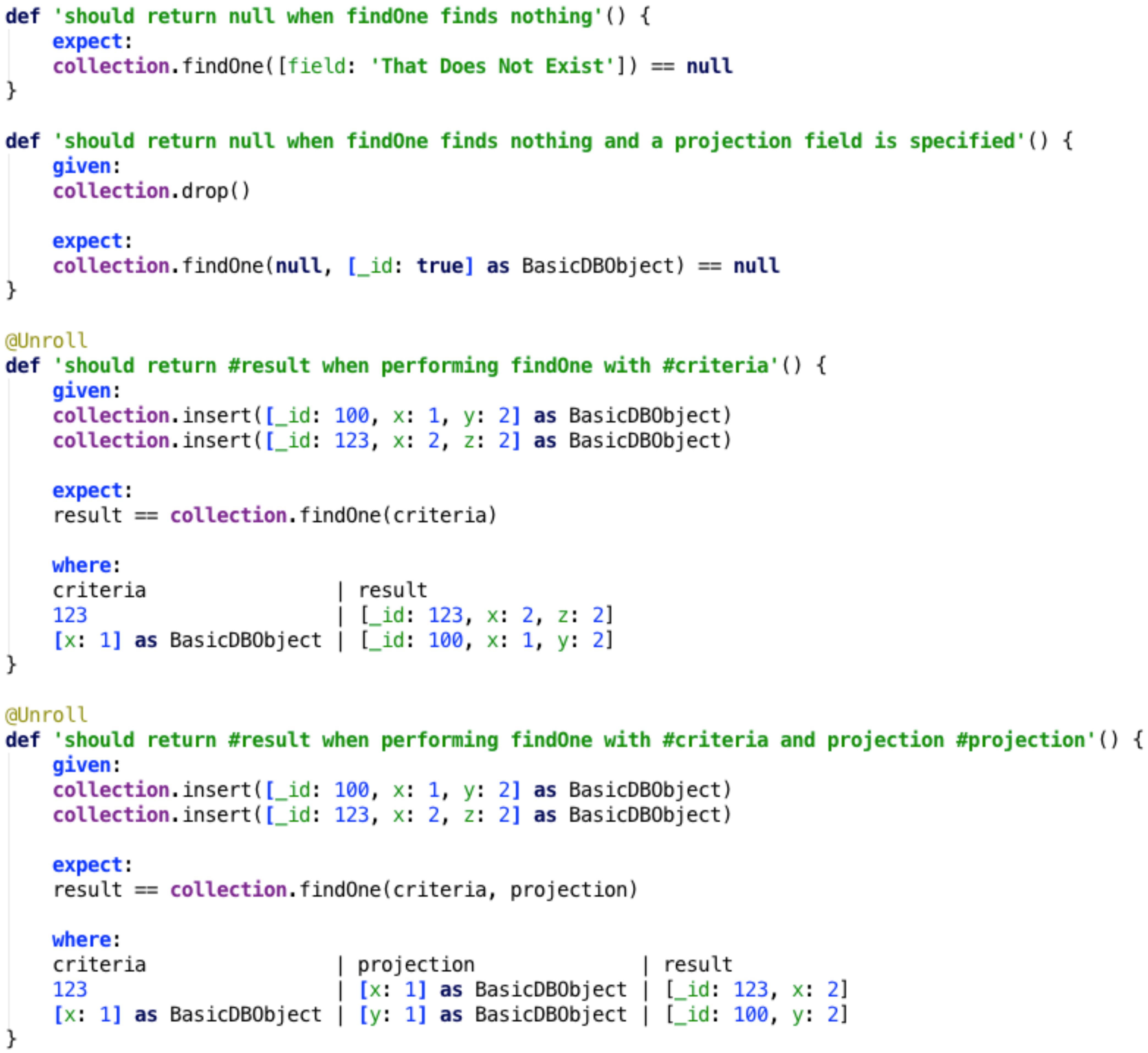
- Mixed team used to different programming styles
- Releases are painful
- Databases schema changes take ages
- Testing is hard and inconsistent



```
@Test
public void testFindOne() {
    DBCollection c = collection;

    DBObject obj = c.findOne();
    assertEquals(obj, null);

    obj = c.findOne();
    assertEquals(obj, null);

    obj = c.findOne();
    assertEquals(obj, null);

    // Test that findOne works when fields is specified but no match is found
    // *** This is a Regression test for JAVA-411 ***
    obj = c.findOne(null, new BasicDBObject("_id", true));

    assertEquals(obj, null);

    DBObject inserted = BasicDBObjectBuilder.start().add("x", 1).add("y", 2).get();
    c.insert(inserted);
    c.insert(BasicDBObjectBuilder.start().add("_id", 123).add("x", 2).add("z", 2).get());

    obj = c.findOne(123);
    assertEquals(obj.get("_id"), 123);
    assertEquals(obj.get("x"), 2);
    assertEquals(obj.get("z"), 2);

    obj = c.findOne(123, new BasicDBObject("x", 1));
    assertEquals(obj.get("_id"), 123);
    assertEquals(obj.get("x"), 2);
    assertEquals(obj.containsField("z"), false);

    obj = c.findOne(new BasicDBObject("x", 1));
    assertEquals(obj.get("x"), 1);
    assertEquals(obj.get("y"), 2);

    obj = c.findOne(new BasicDBObject("x", 1), new BasicDBObject("y", 1));
    assertEquals(obj.containsField("x"), false);
    assertEquals(obj.get("y"), 2);
}
```

Now What?

- Identify possible technology / process options, e.g:
  - Java 8, Groovy
  - DevOps
  - NoSQL
  - Modern Test Frameworks (e.g. Spock, Lambda Behave)



```
def 'should return null when findOne finds nothing'() {
    expect:
    collection.findOne([field: 'That Does Not Exist']) == null
}

def 'should return null when findOne finds nothing and a projection field is specified'() {
    given:
    collection.drop()

    expect:
    collection.findOne(null, [_id: true] as BasicDBObject) == null
}

@Unroll
def 'should return #result when performing findOne with #criteria'() {
    given:
    collection.insert([_id: 100, x: 1, y: 2] as BasicDBObject)
    collection.insert([_id: 123, x: 2, z: 2] as BasicDBObject)

    expect:
    result == collection.findOne(criteria)

    where:
    criteria                   | result
    123                        | [_id: 123, x: 2, z: 2]
    [x: 1] as BasicDBObject | [_id: 100, x: 1, y: 2]
}

@Unroll
def 'should return #result when performing findOne with #criteria and projection #projection'() {
    given:
    collection.insert([_id: 100, x: 1, y: 2] as BasicDBObject)
    collection.insert([_id: 123, x: 2, z: 2] as BasicDBObject)

    expect:
    result == collection.findOne(criteria, projection)

    where:
    criteria                   | projection                 | result
    123                        | [x: 1] as BasicDBObject | [_id: 123, x: 2]
    [x: 1] as BasicDBObject | [y: 1] as BasicDBObject | [_id: 100, y: 2]
}
```

Approaches

- Spikes / prototypes
- Use in tests
- Use in non-production code
  - Build / Deploy / Continuous Delivery Pipeline
- Use in internal tools
- Skunkworks projects / hack days

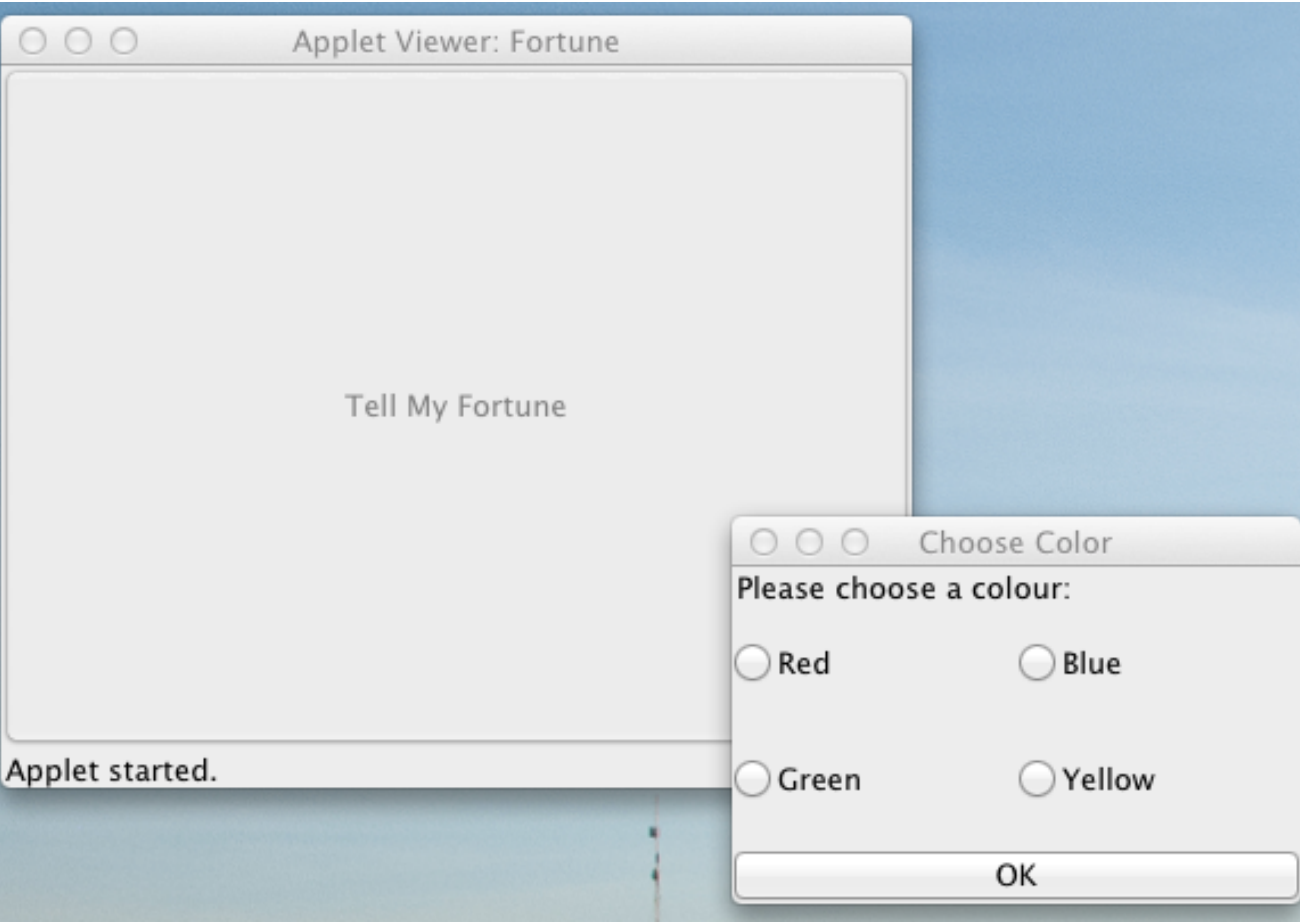
For You

Why?

- I want to have fun
- I want a new job
- I want to be a Technology Guru in this job
- I don’t want to be left behind

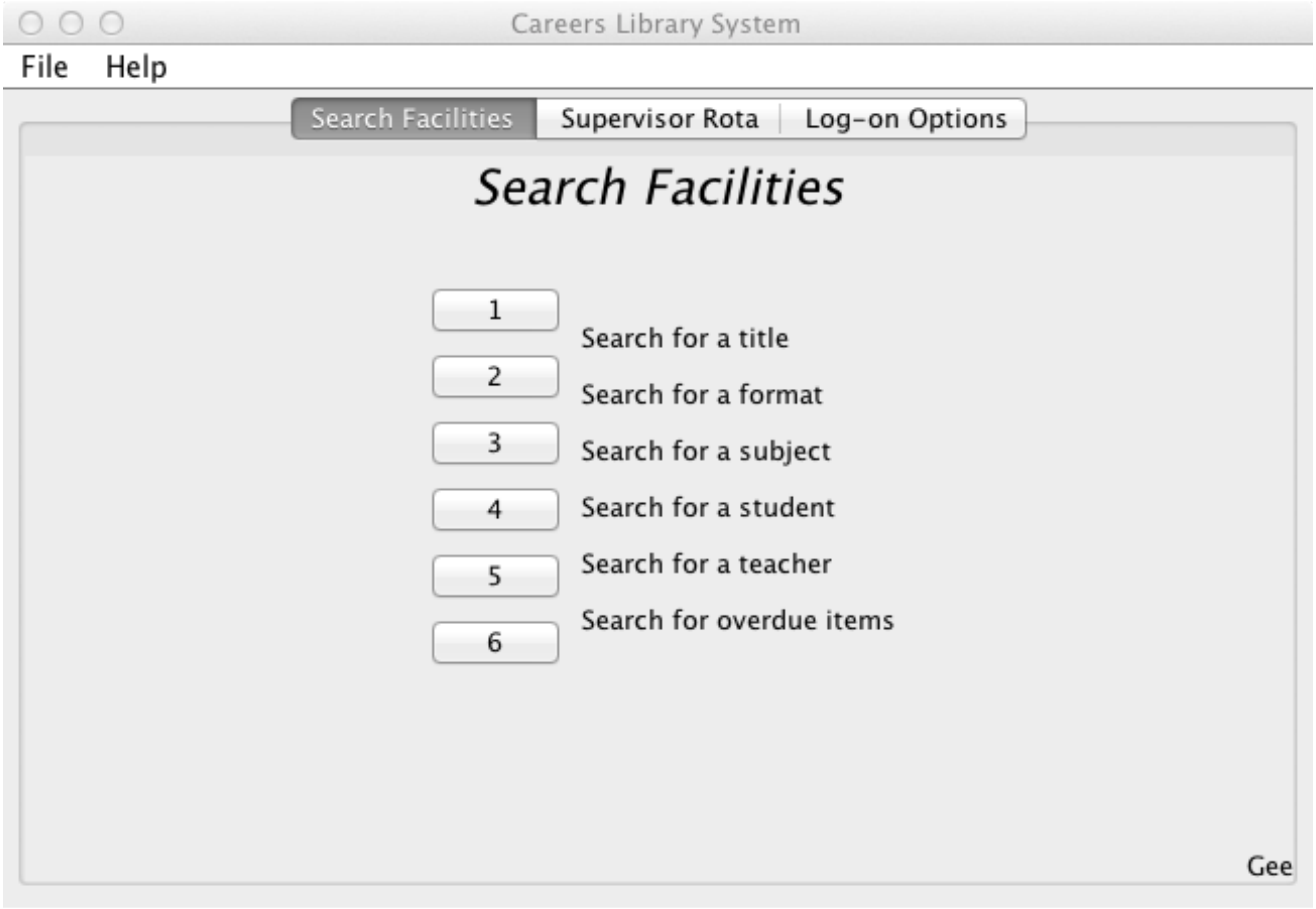
Now What?

- Any technology that looks awesome
- Buzz words from job adverts
- Any technology that is currently (or soon to be) used at work
- Any/all of the above. Or None.



Please choose a colour:

Applet started.


Careers Library System
File
Help
Search Facilities
Supervisor Rota
Log-on Options
Search Facilities
1
Search for a title
2
Search for a format
3
Search for a subject
4
Search for a student
5
Search for a teacher
6
Search for overdue items
Gee

Approaches

- User Groups
- Online courses
- Pet project
- Join an open source project
- Find a project at work
- Get a new job!

Why?

- Because you want to

Being Proactive

Preventing Premature Implementation

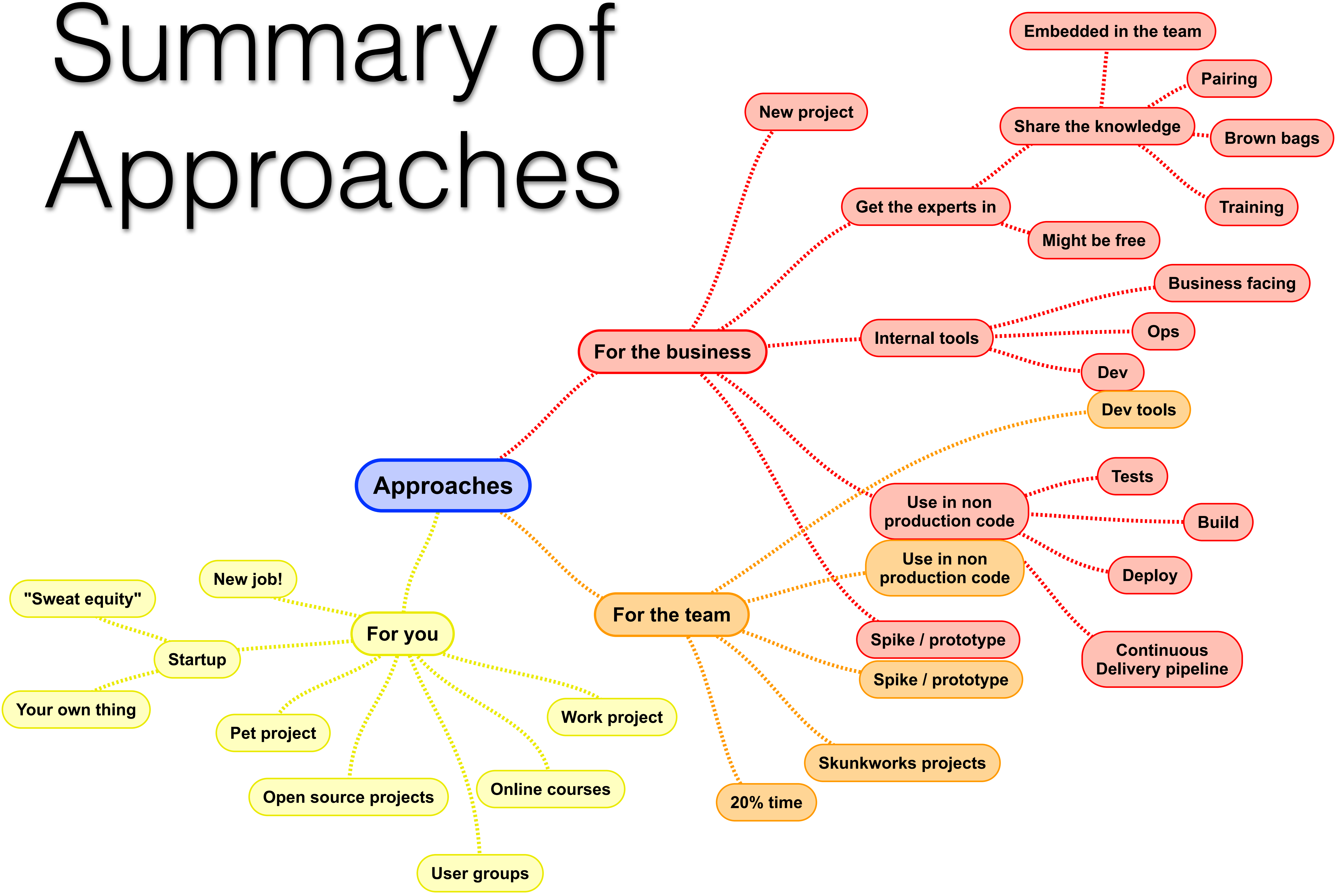
- Hack days
- Skunkworks
- 20% time
- Conferences
- Brown Bags
- User Groups

Risks

Benefits

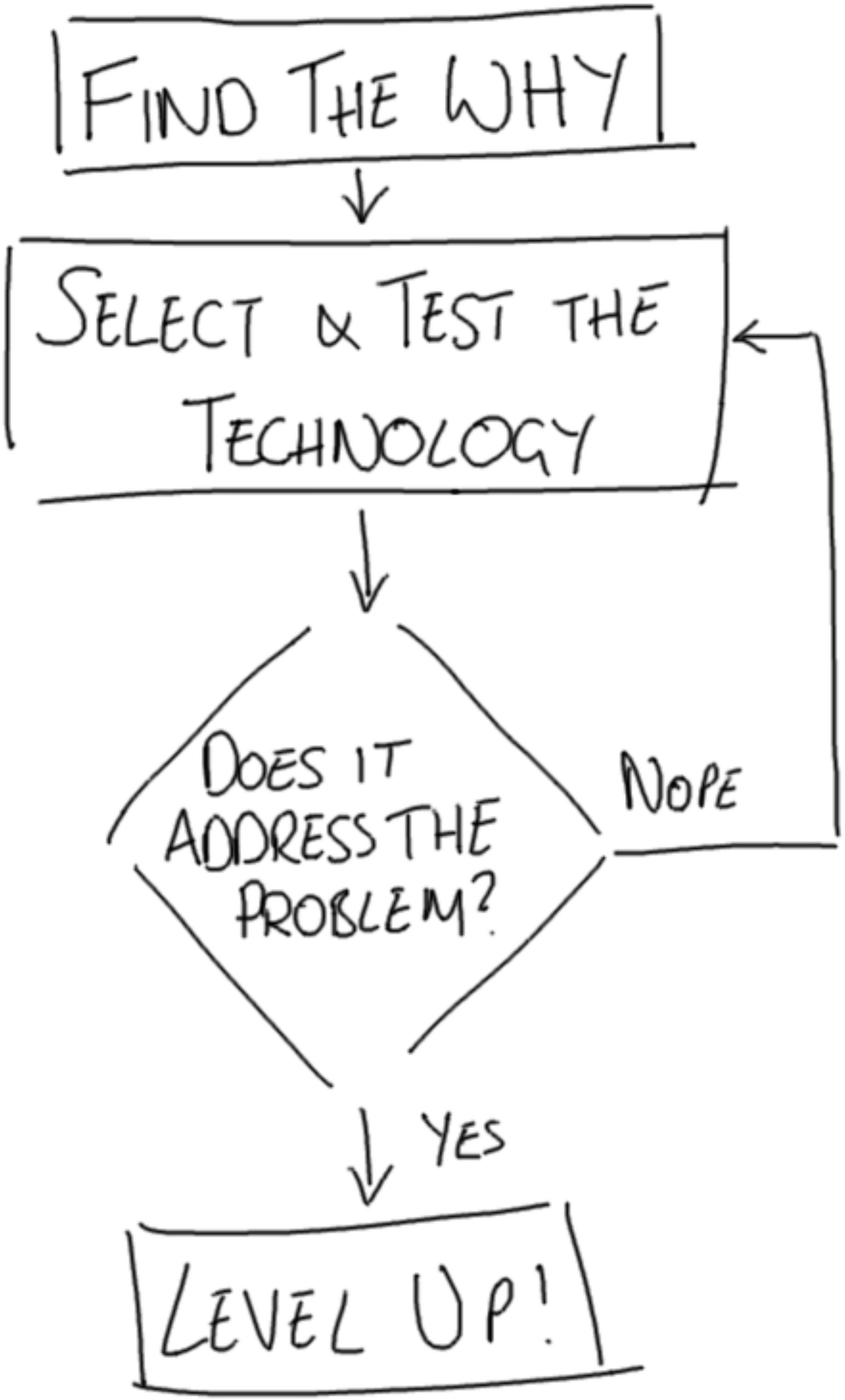
- Skilled Developers
- Ready to adopt the right technology at the right time
- Happy Developers

Summary of Approaches



So What Can I Do?

Don’t Panic!


FIND THE WHY
SELECT & TEST THE TECHNOLOGY
DOES IT ADDRESS THE PROBLEM?
NOPE
YES
LEVEL UP!

Lambdas: The Right Way

http://bit.ly/
stayingahead