

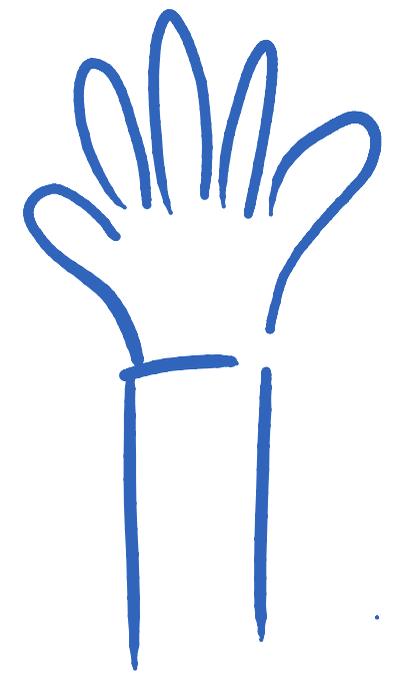
BDD

to the point 

Nikolas.M@rtens.org

bephpug, 7.5.2013

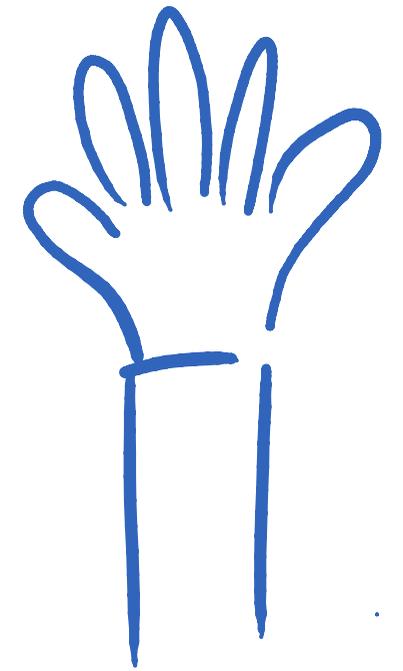
Show some hands



keep
'em up!

Show some hands

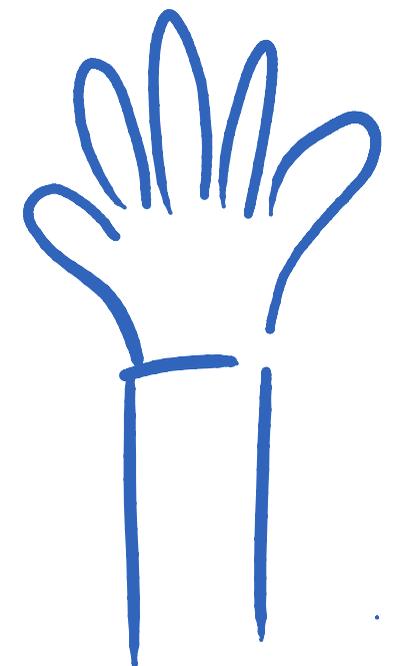
Who... heard of it?



keep
'em up!

Show some hands

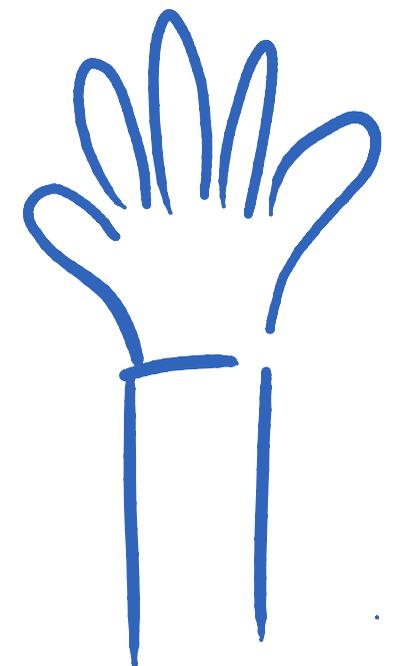
Who... heard of it?
uses it?



keep
'em up!

Show some hands

Who... heard of it?
uses it?
daily?



keep
'em up!

What the hash ~~is~~ is BDD



What the h#  k is



BDD

0 5 8 > 0 1

5 0 < 1 7

4 5 0 3 7 0 - 0 2 3 0 5 4

What the h#  k is BDD

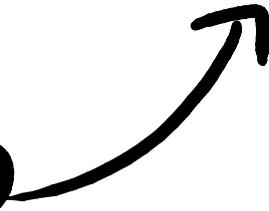


0 < 1 < 2 < 3 < 4

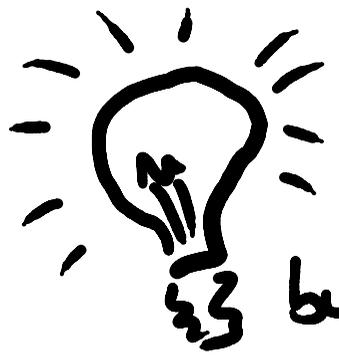
5 < 6 < 7

8 < 9 < 10 < 11 < 12

Like TDD
but with B



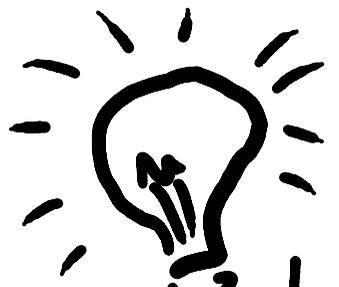
What the h#k is BDD



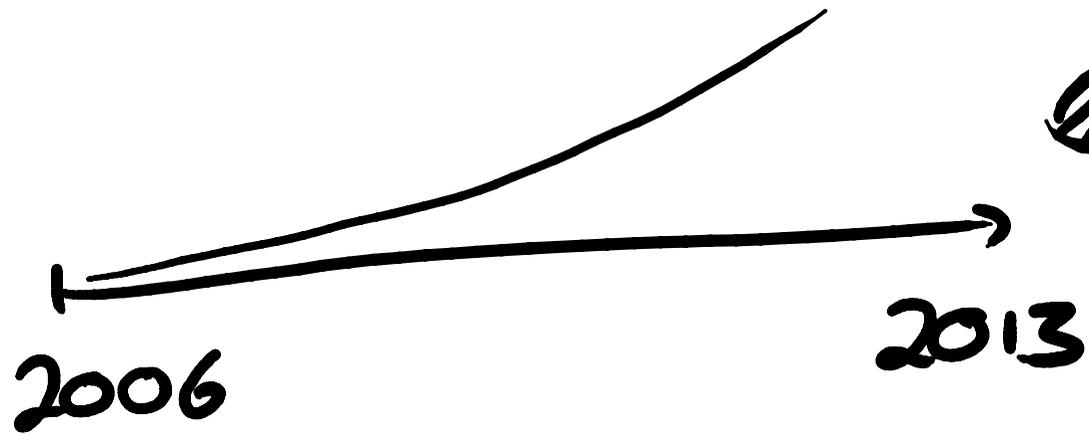
by Dan North



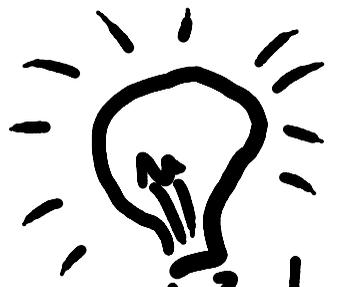
What the h#k is BDD



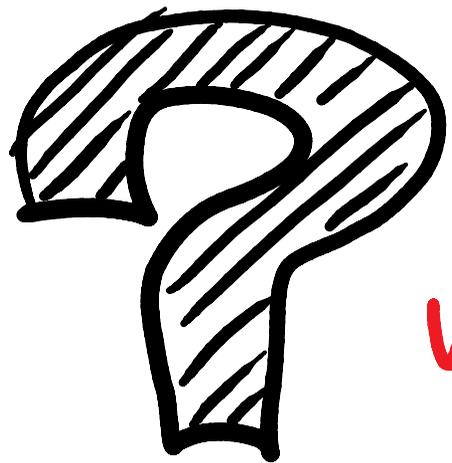
by Dan North



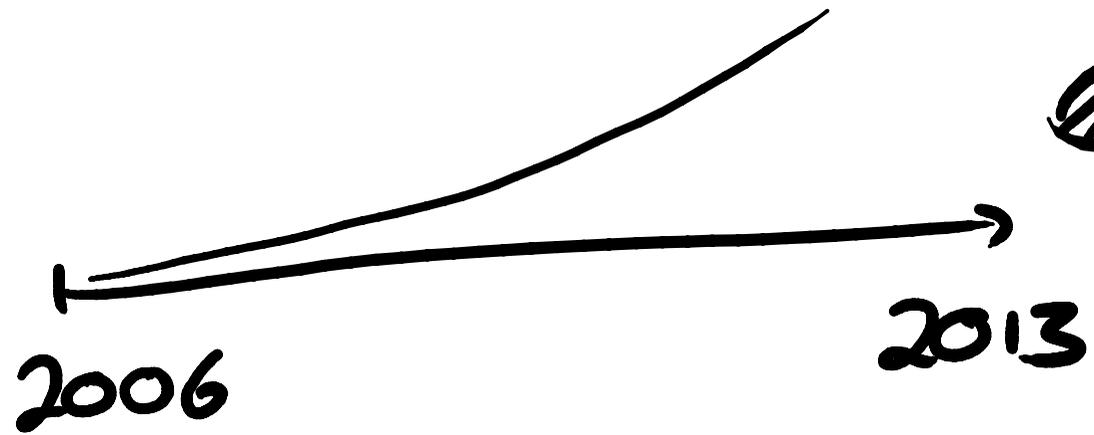
What the h#k is BDD



by Dan North



But now?



What is this talk about

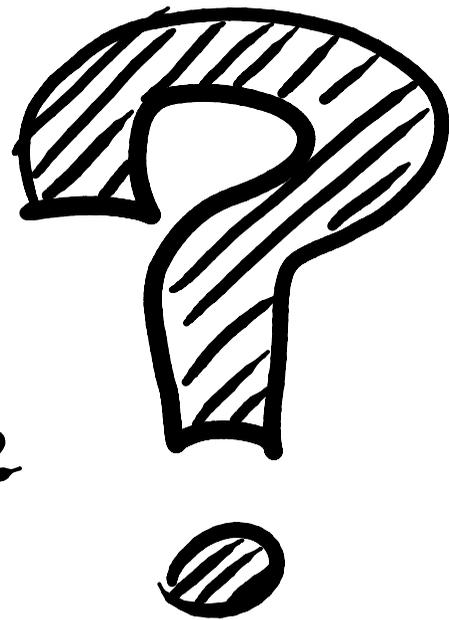


What is this talk about

My view

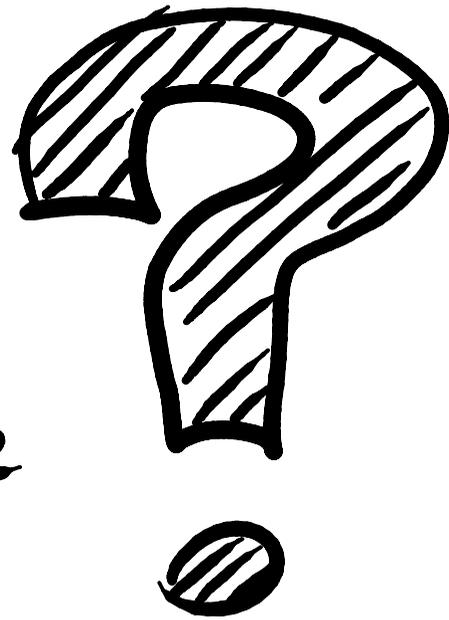
My experience

How I use it



What is this talk about

My view
My experience
How I use it



To the
point



no libraries
no frameworks
no tools

MD

MM@

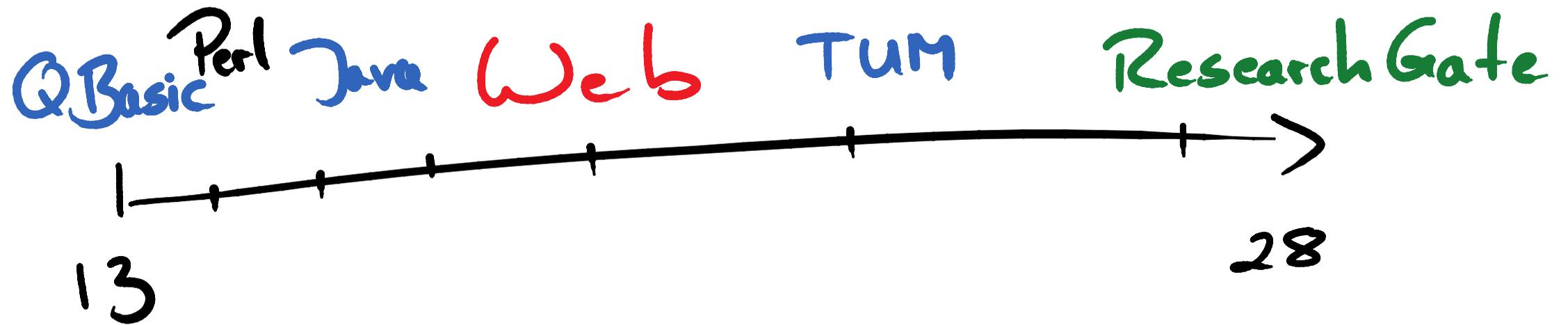
Nikolas.M@rtens.org

<http://rtens.org>

MM@

Nikolas.M@rtens.org

<http://rtens.org>



What is BDD about?

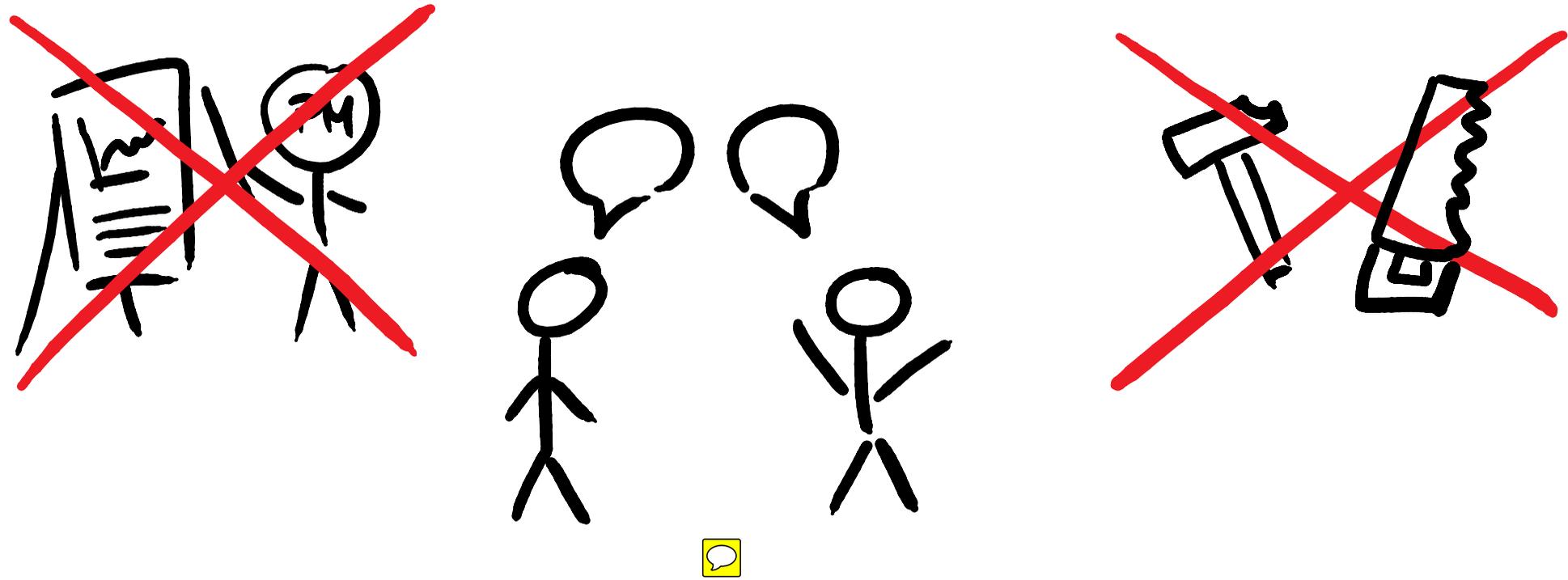
What is BDD about?



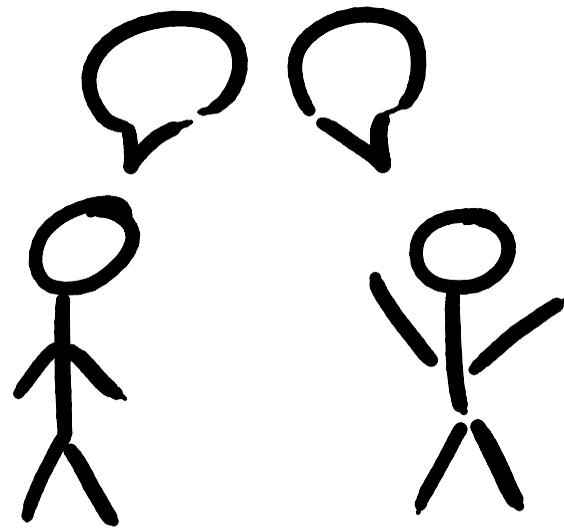
What is BDD about?



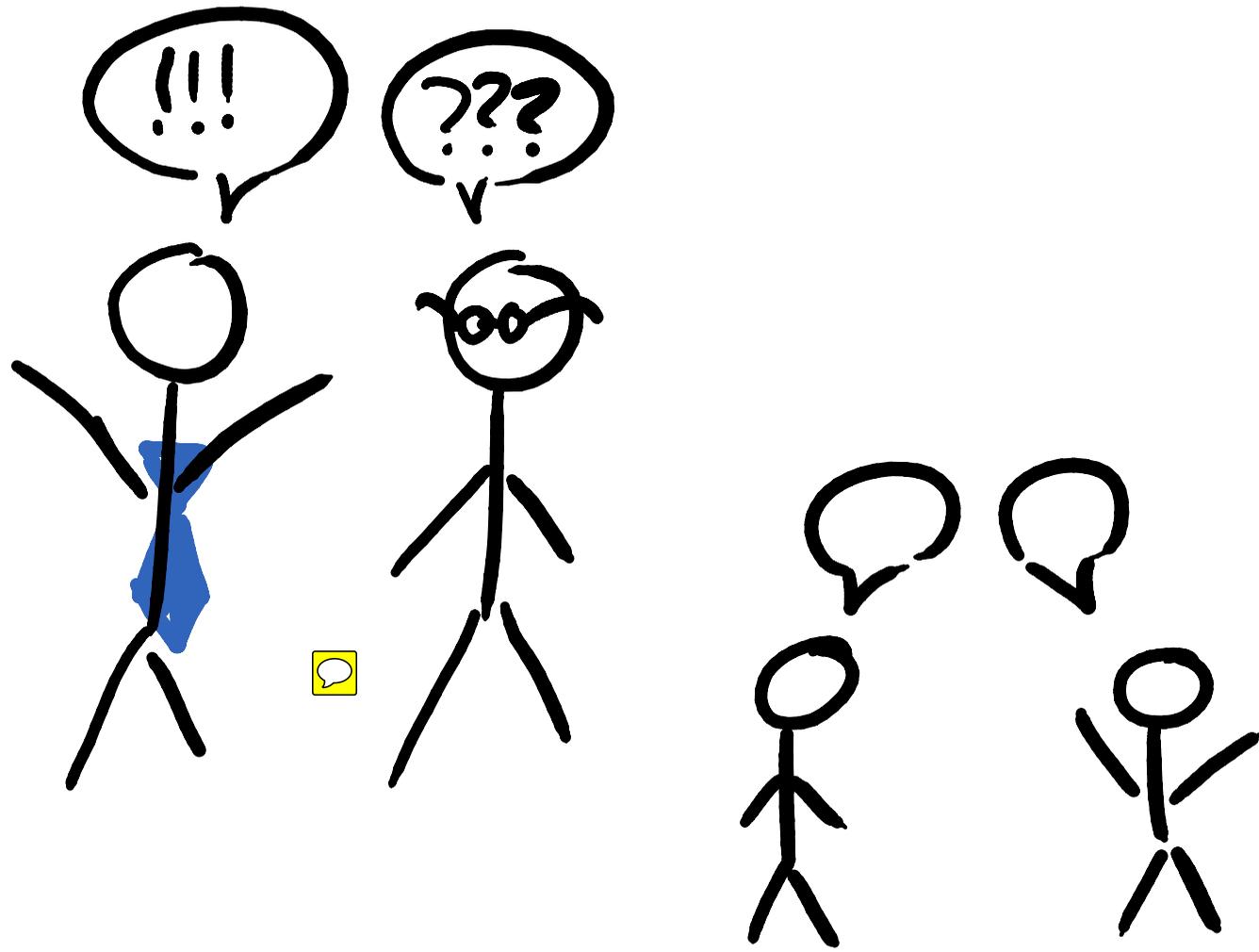
What is BDD about?



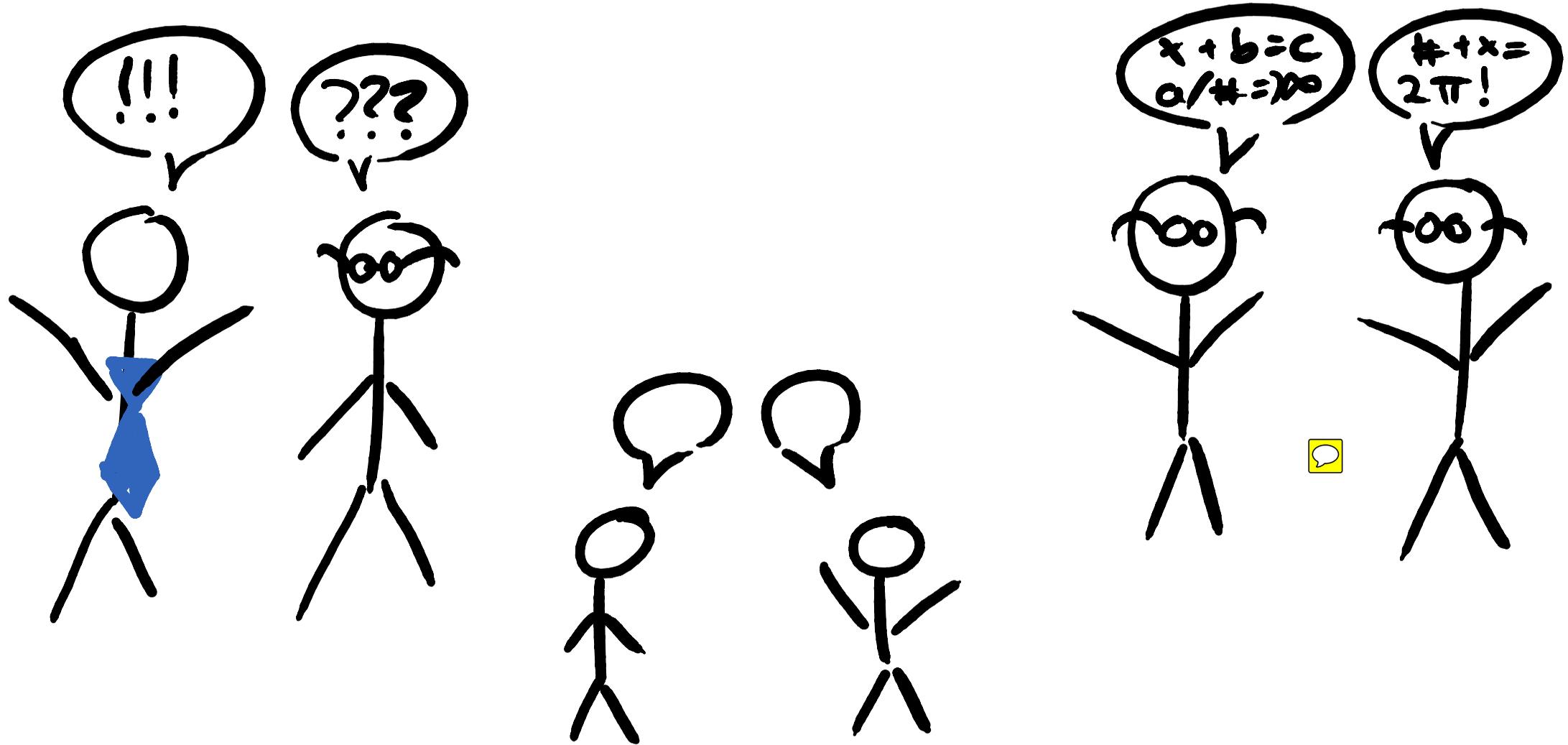
Communication



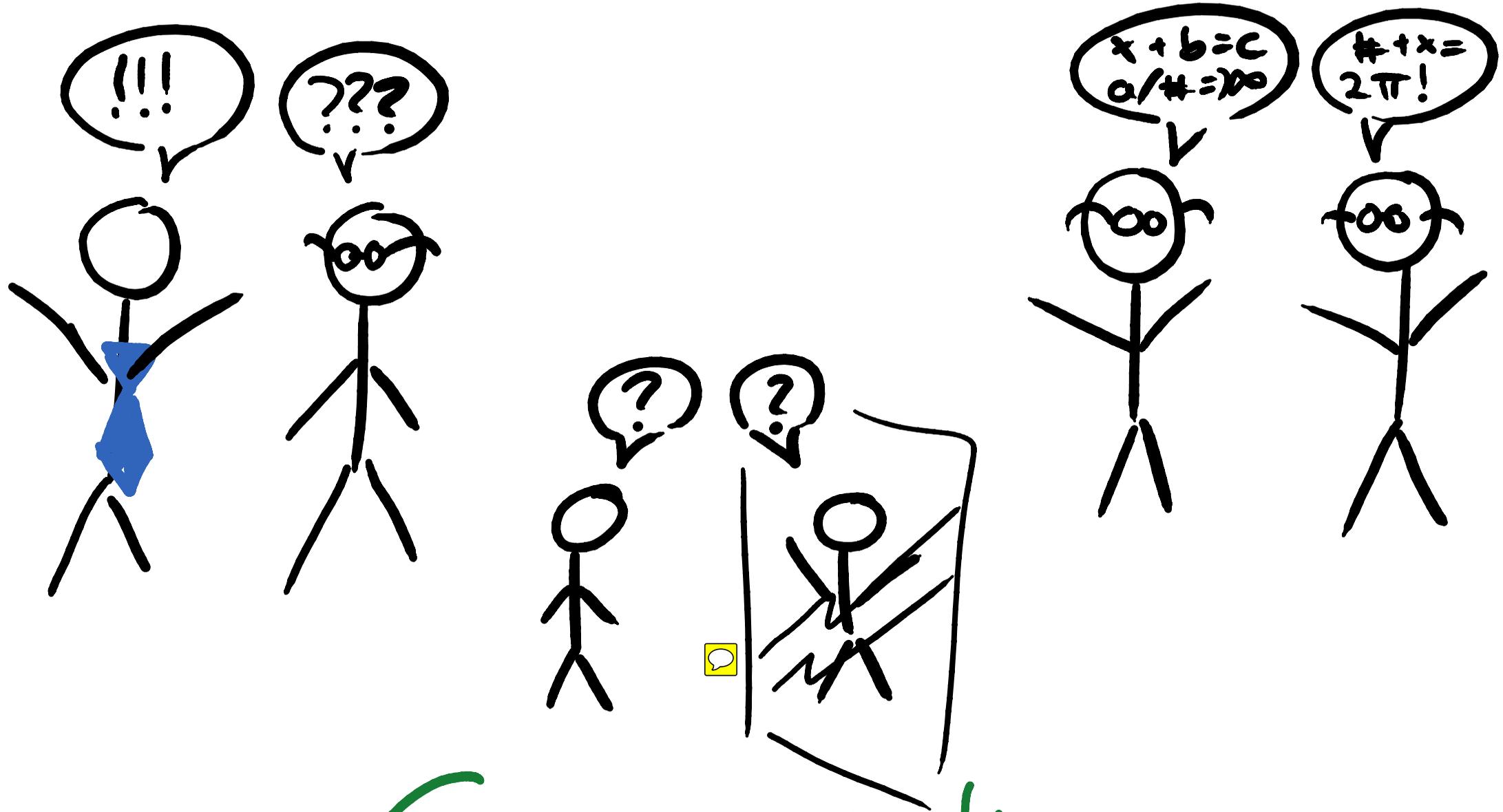
Communication



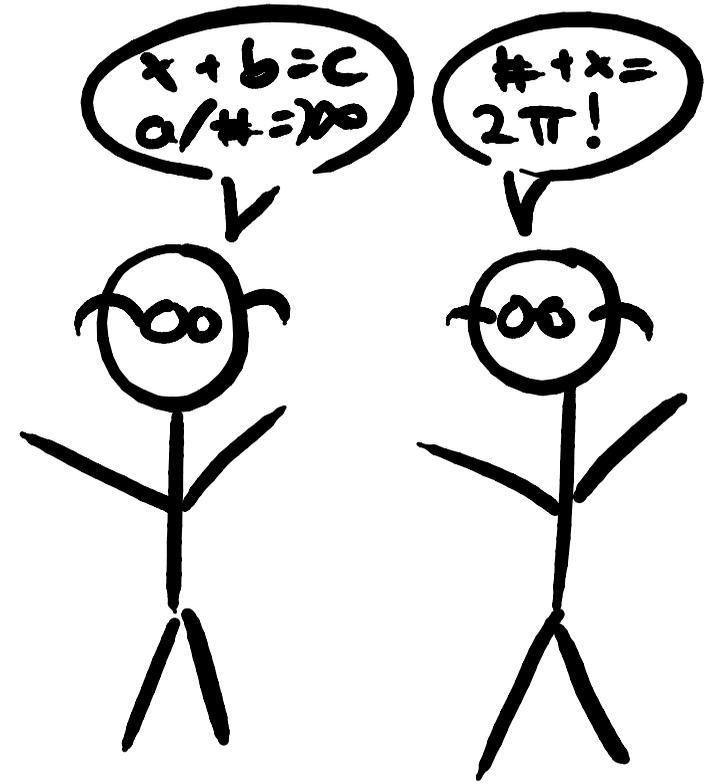
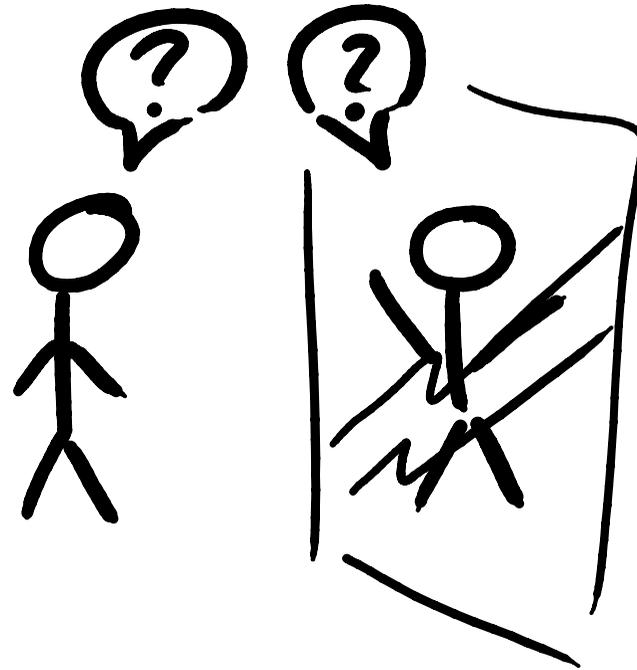
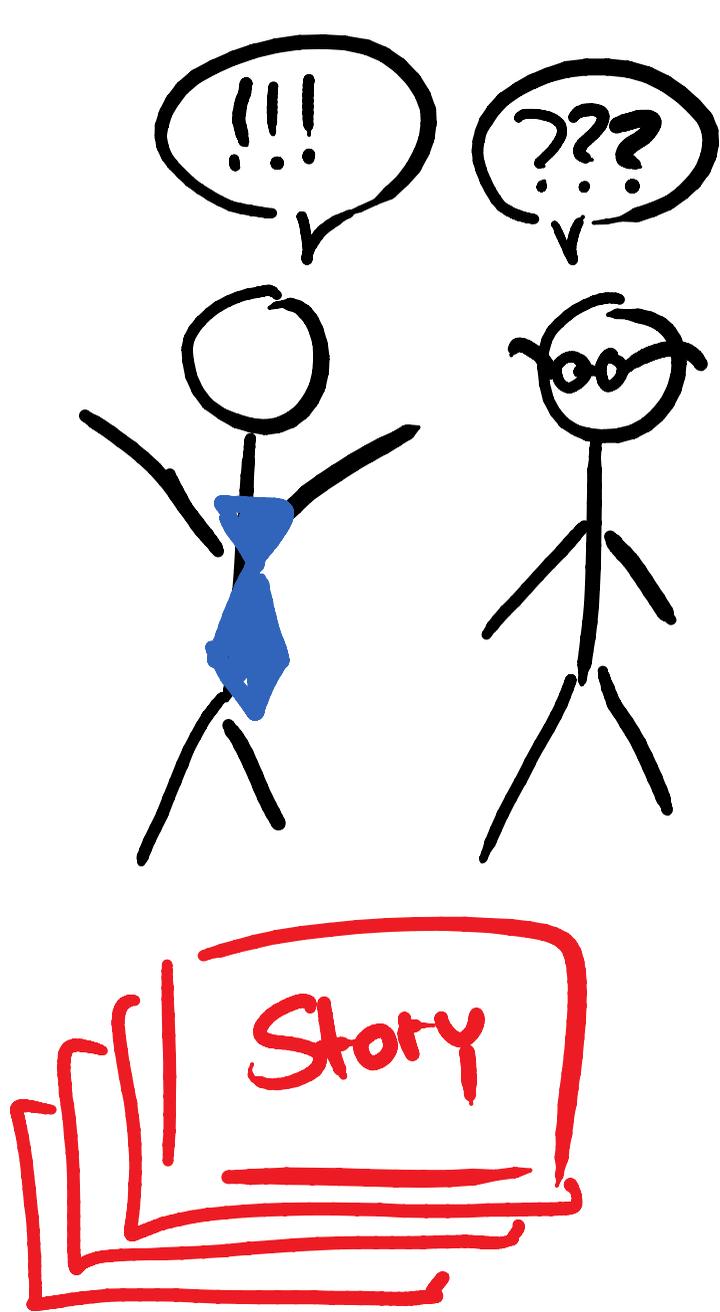
Communication

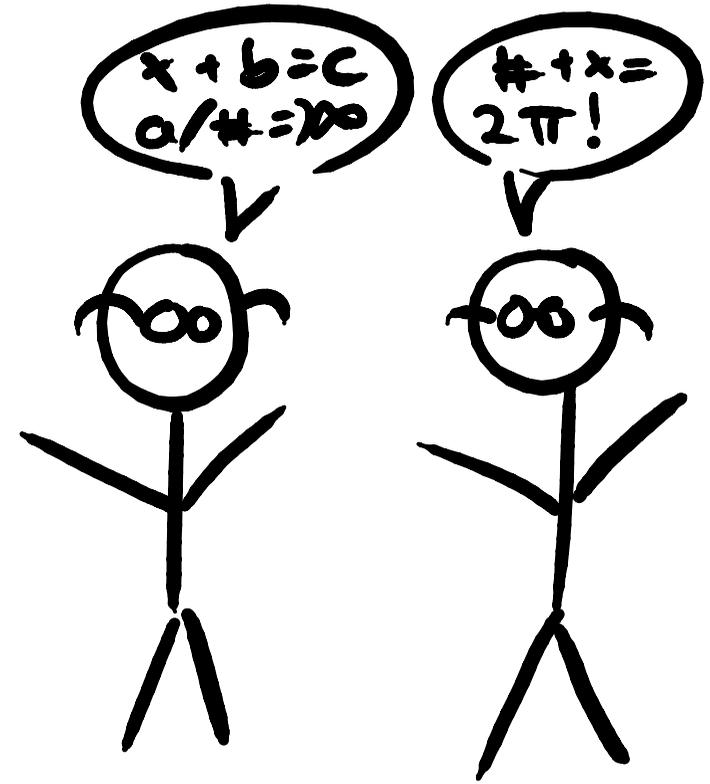
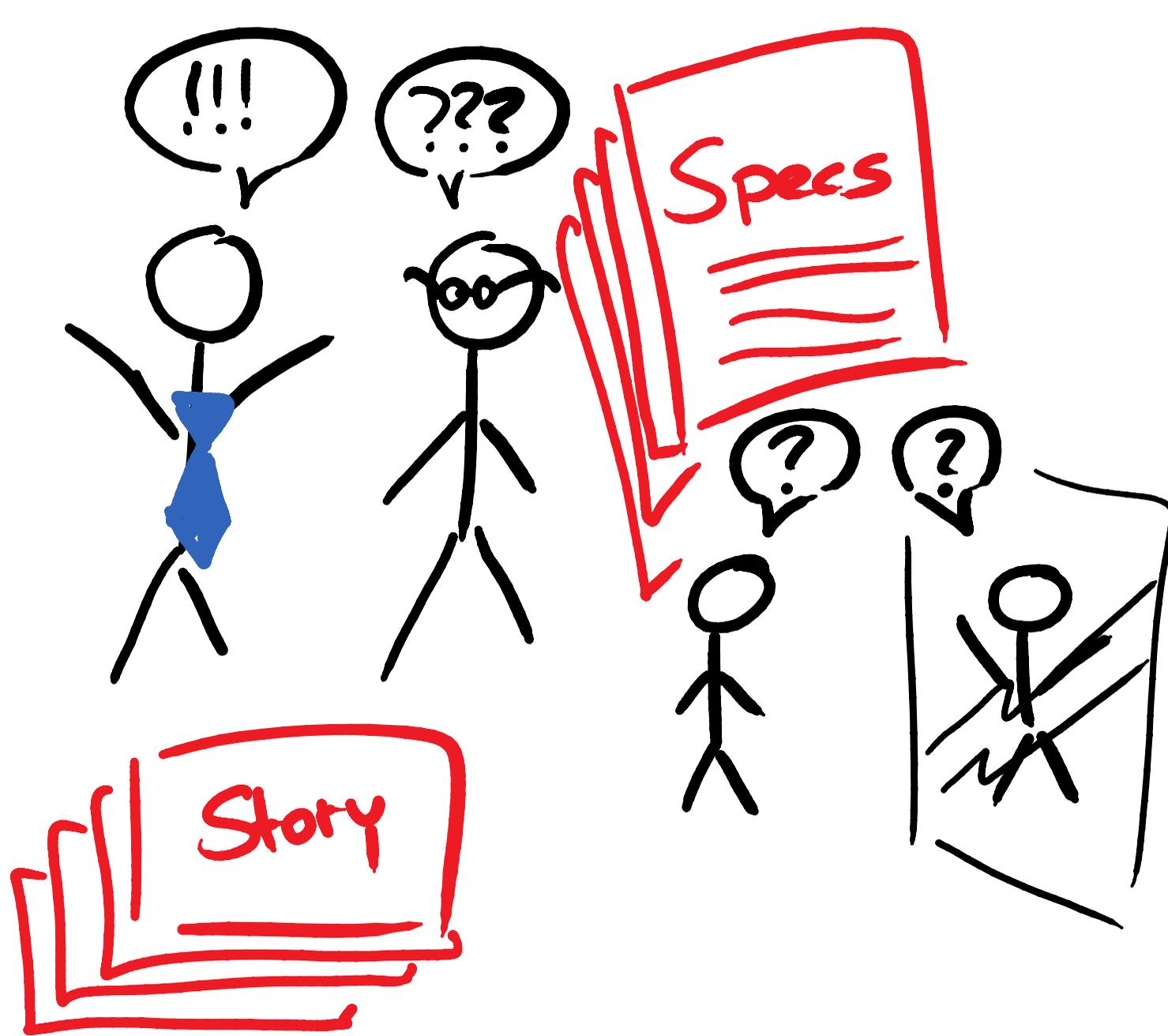


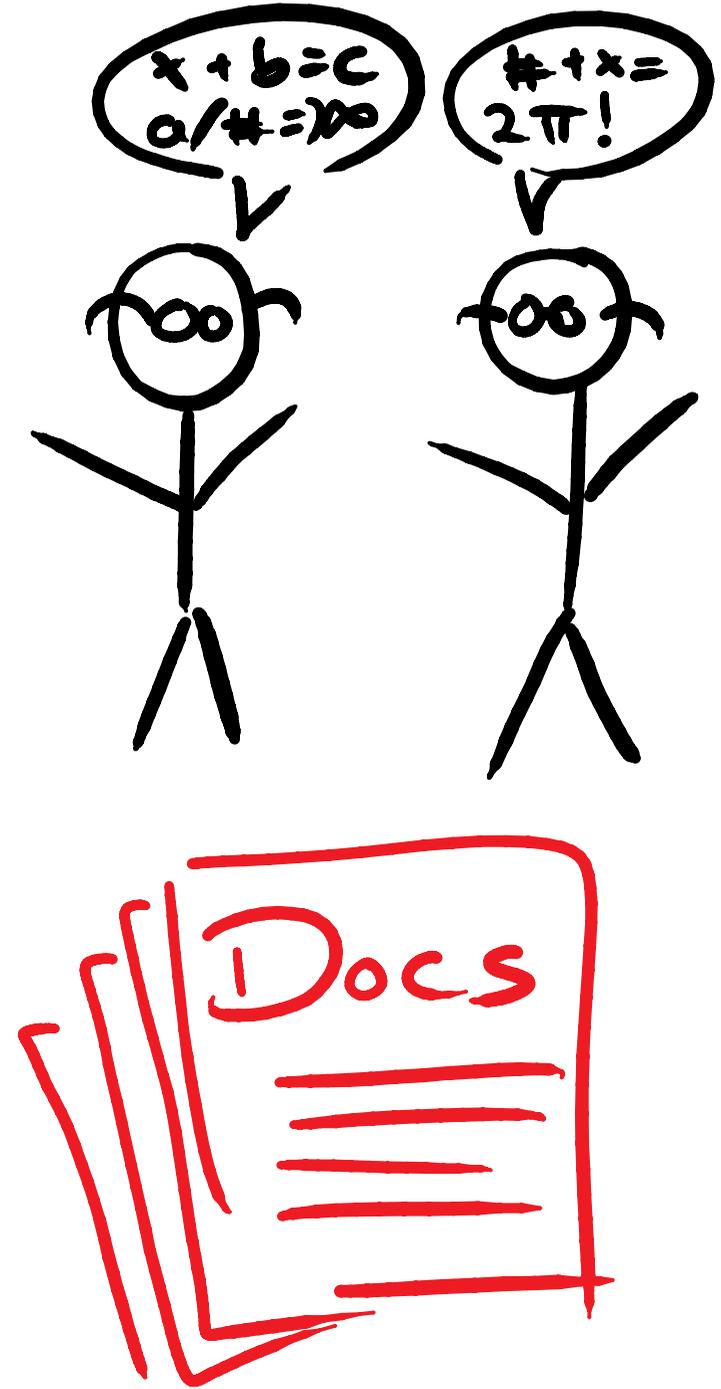
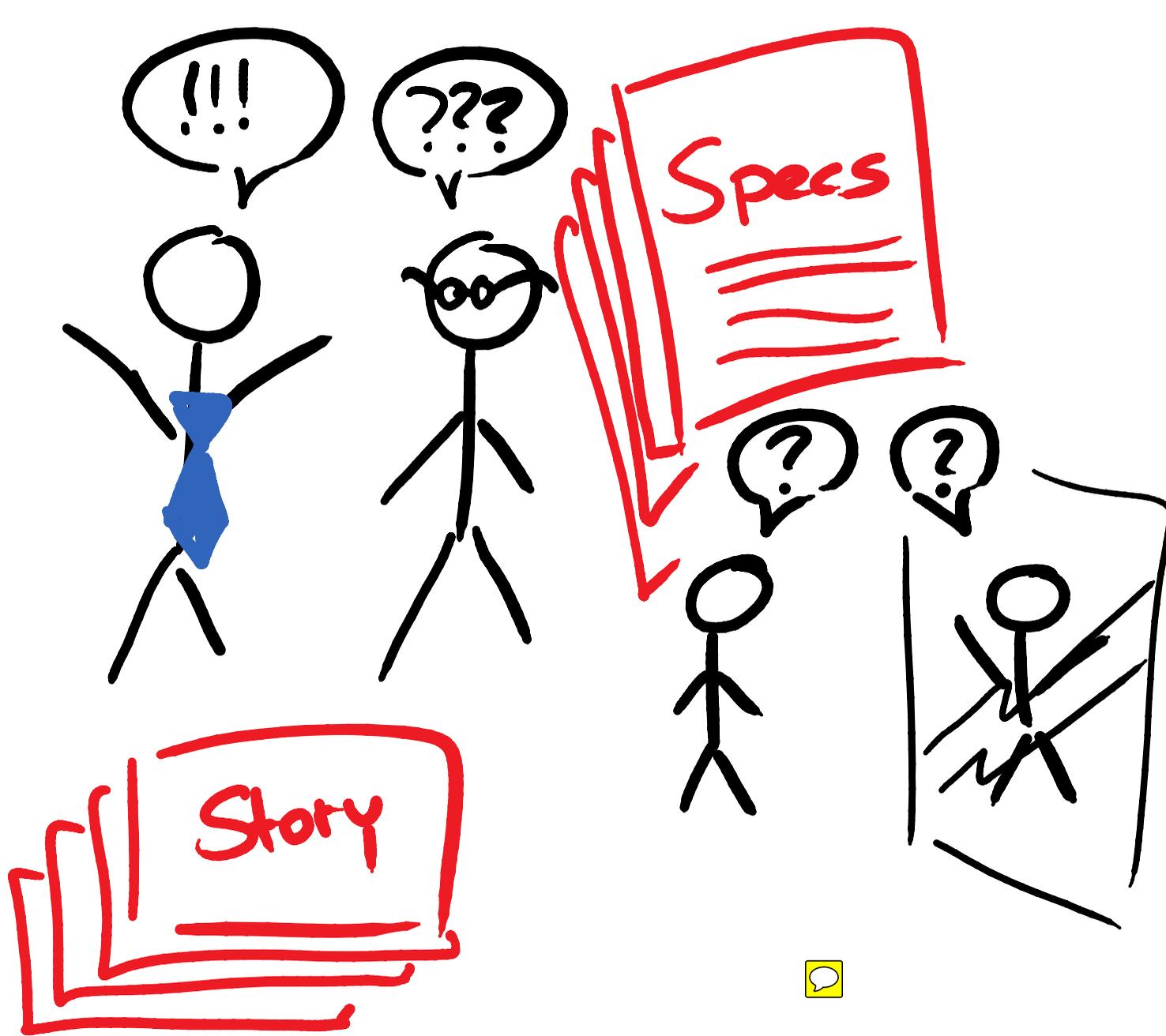
Communication



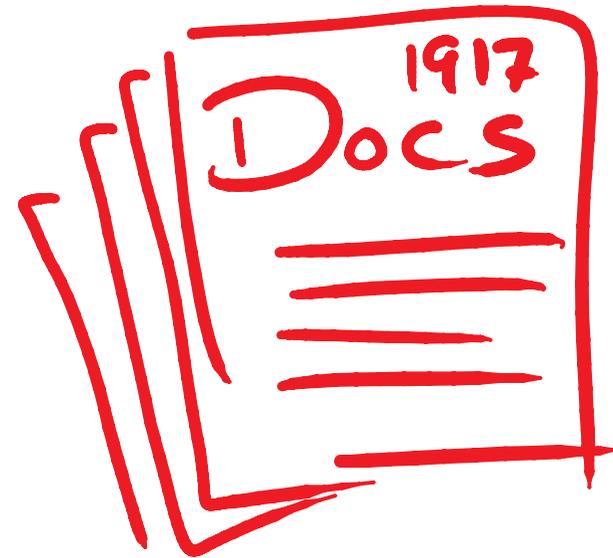
Communication





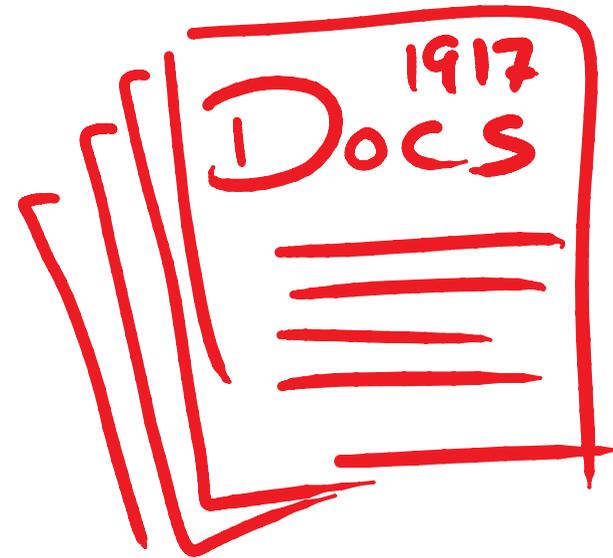


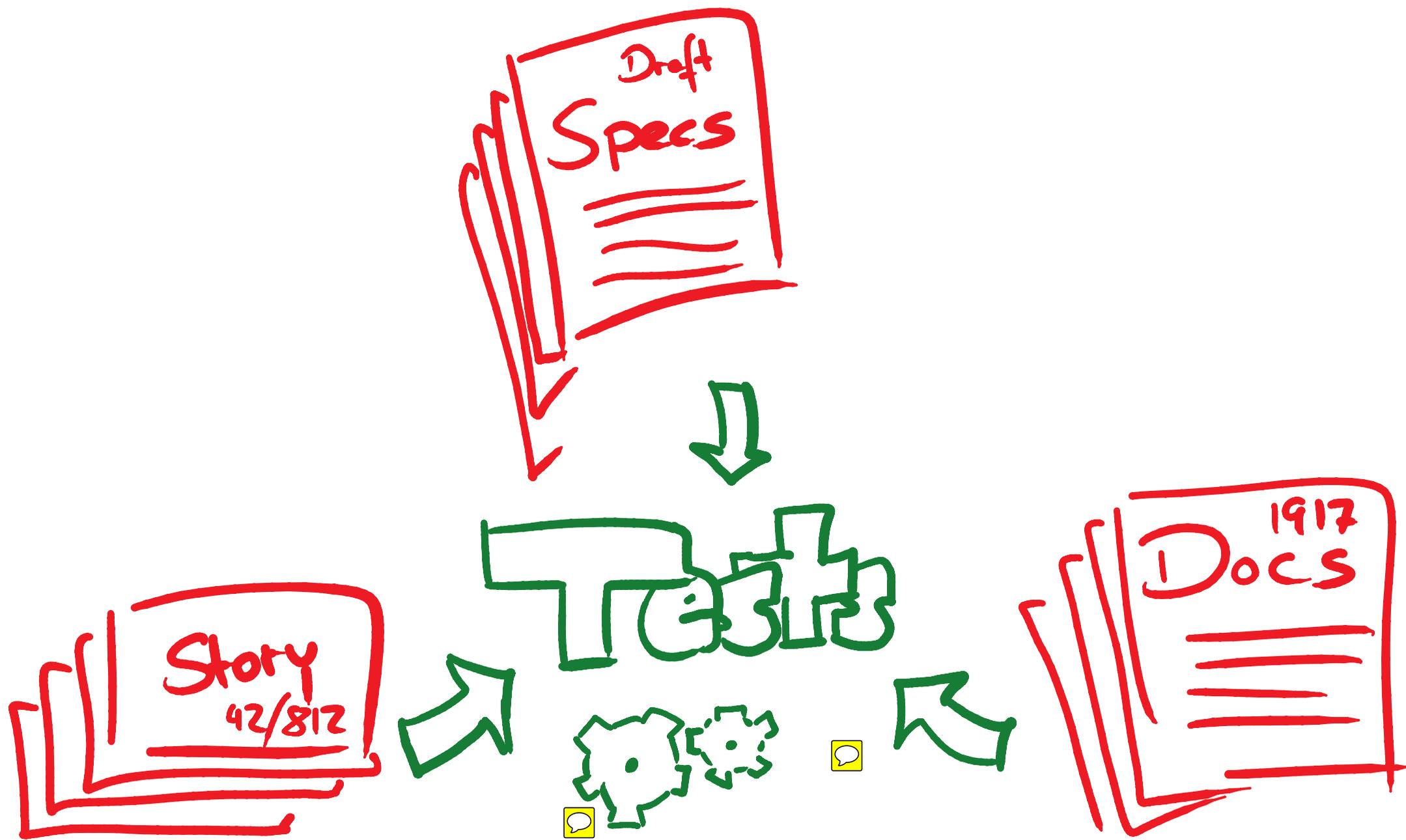






Tests

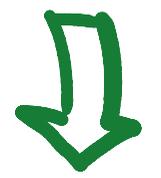




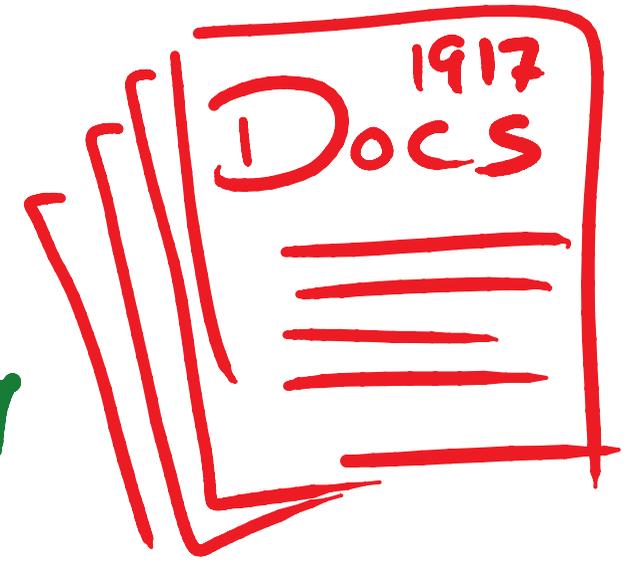
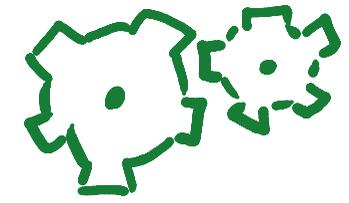
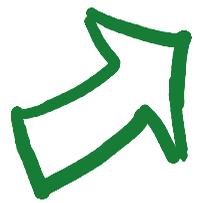


~~test~~ !
Behavior

A stick figure wearing a blue tie, positioned to the right of the text.



Tests

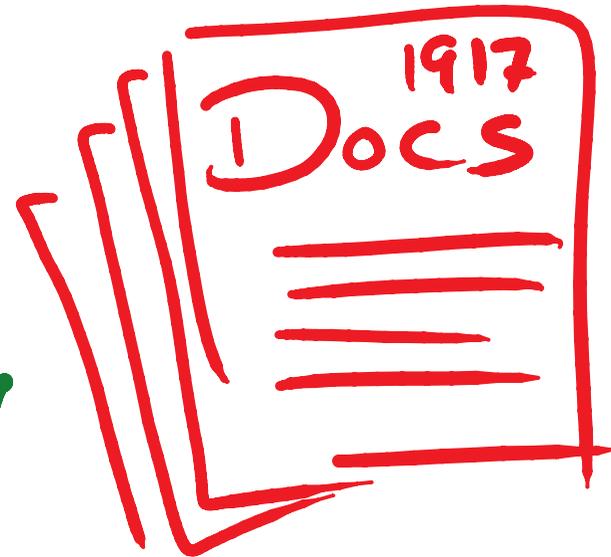
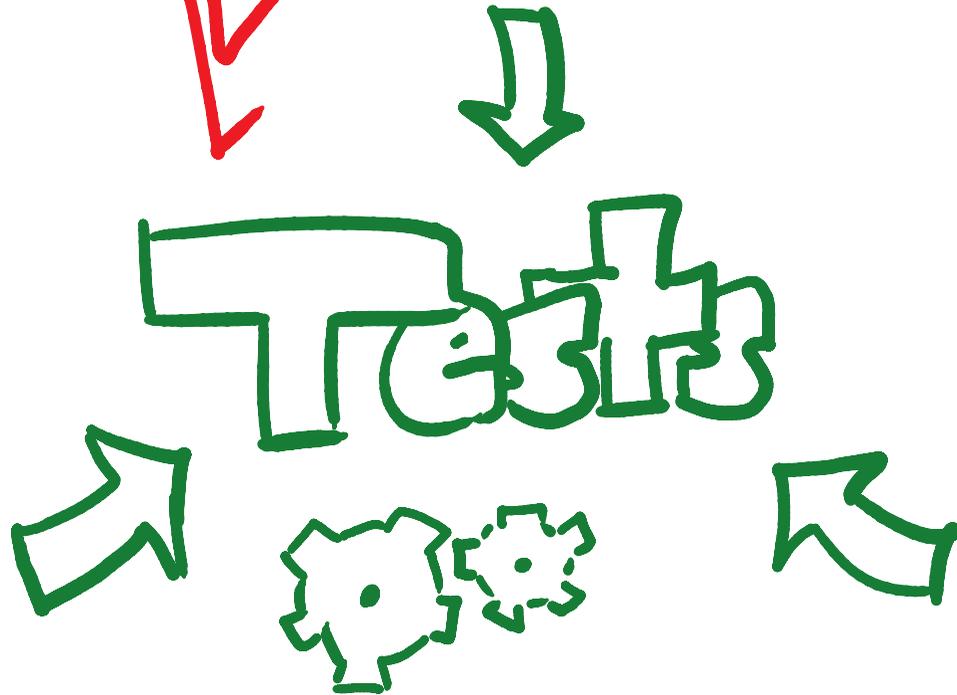


~~PHP~~
~~JAVA~~
~~C#~~

Domain Language 



~~test~~ 
Behavior 

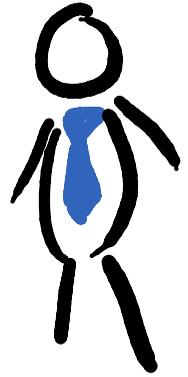


~~PHP~~
~~JAVA~~
~~C#~~

Domain
Language

Draft
Specs

~~test~~

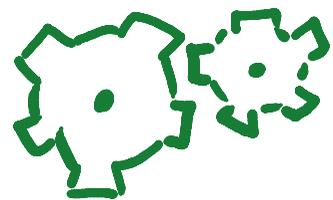


Behavior

BDD

Story
42/812

Implementation



1917
DOCS

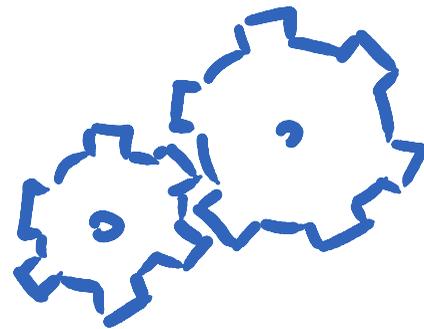
What again?

BDD

Domain language ~~010110011
10011100~~

Behavior ~~test~~

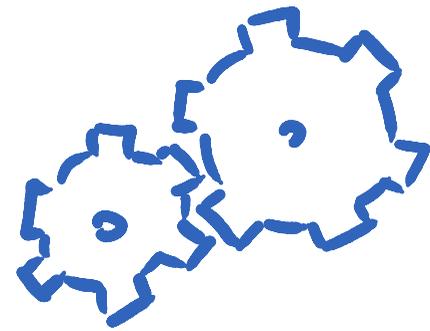
BDD



Executable

Domain language

Behavior

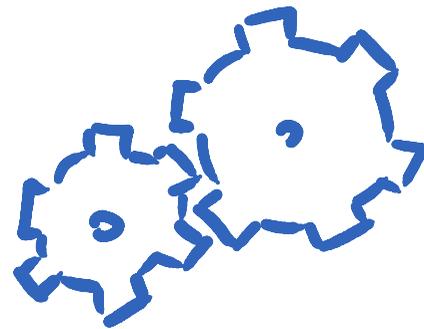


Executable

Domain language

- everybody understands
- clarifies intentions 
- improves discussion

Behavior



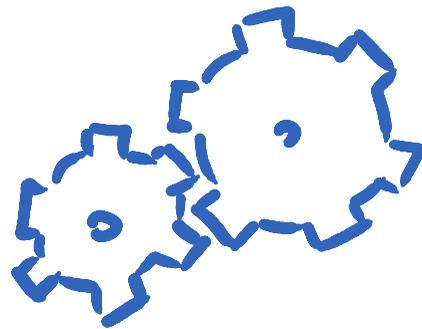
Executable

Domain language

- everybody understands
- clarifies intentions
- improves discussion

Behavior

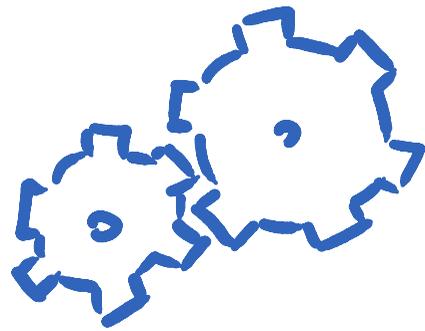
- less technical
- focus on behavior not implementation
- maps business values



Executable

Domain language

- everybody understands
- clarifies intentions
- improves discussion



Executable

Behavior

- less technical
- focus on behavior not implementation
- maps business values
- always up-to-date
- single source of truth
- self-validating

Sounds nice,
but how...?

Sounds nice,
but how...?
With structure!

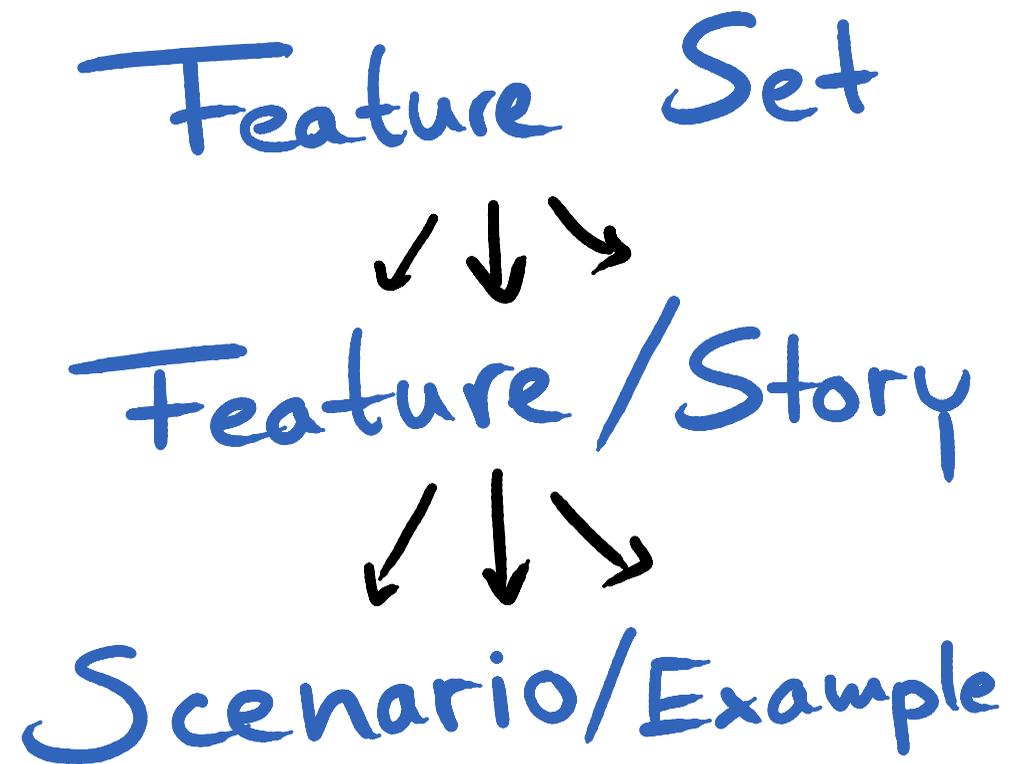
Sounds nice,
but how...?
With structure!

Feature Set

Sounds nice,
but how...?
With structure!

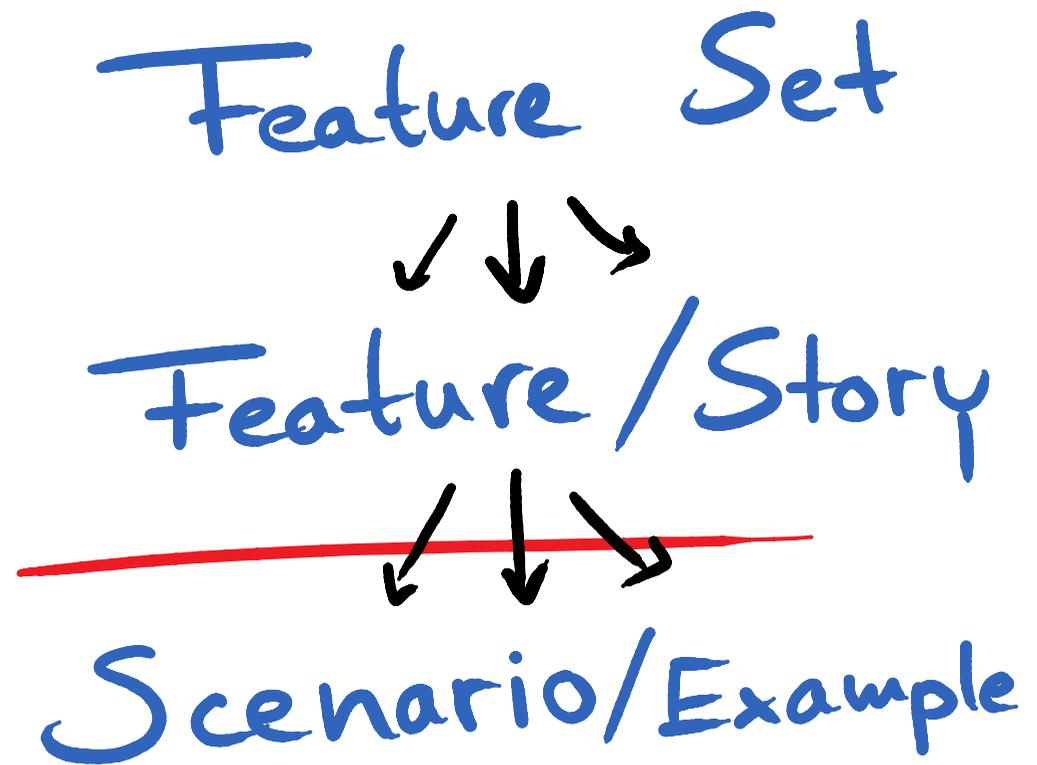
Feature Set
↙ ↓ ↘
Feature / Story

Sounds nice,
but how...?
With structure!



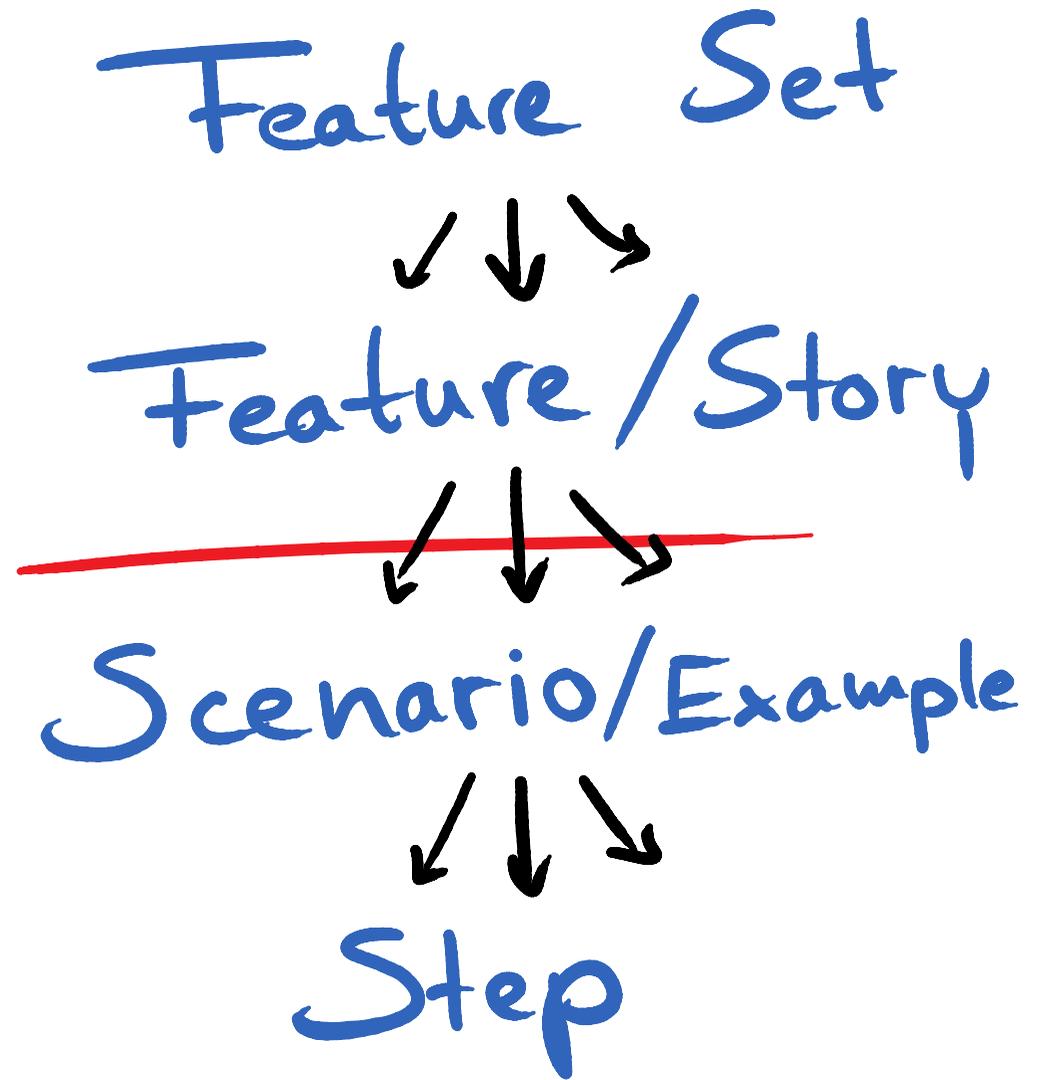
Sounds nice,
but how...?
With structure!

business
cares until
here

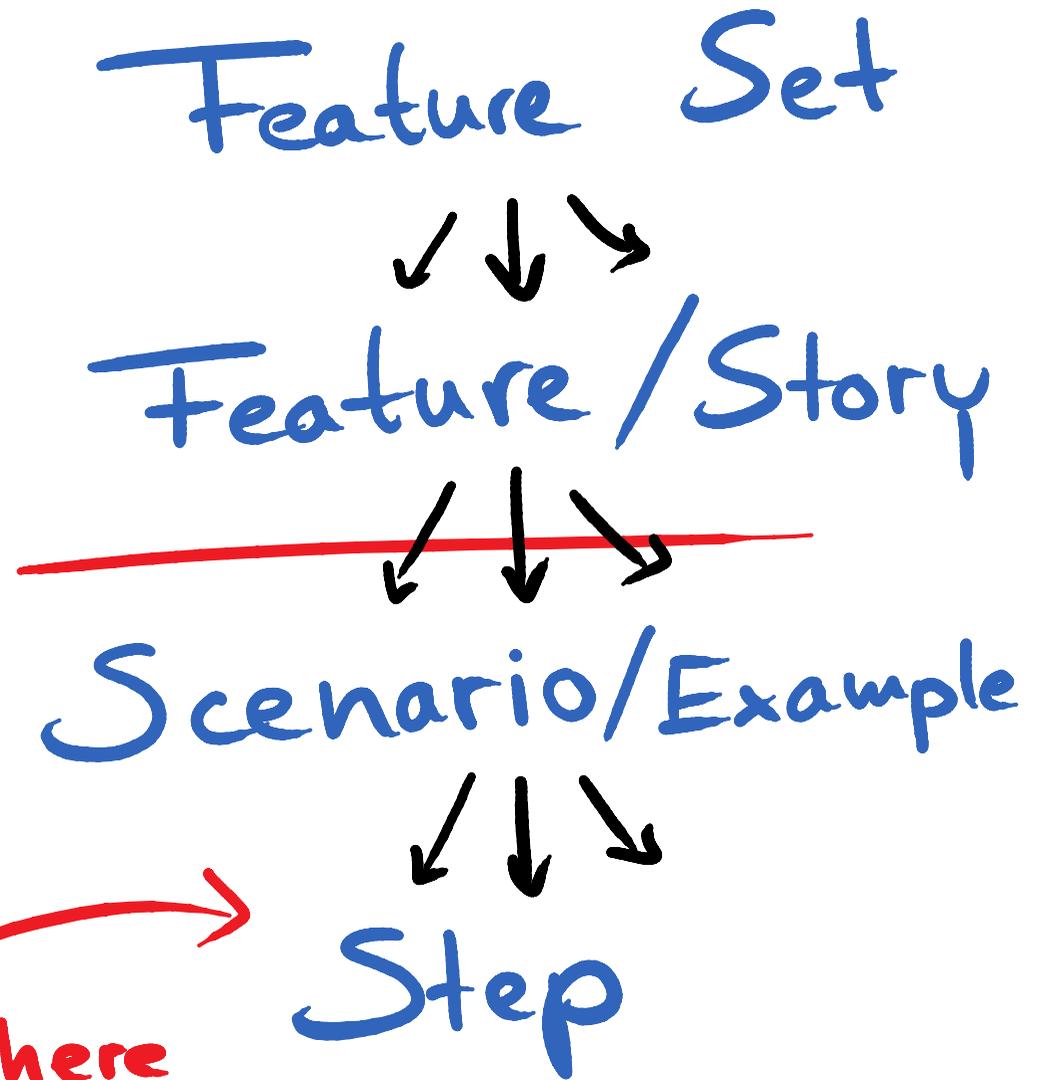


Sounds nice,
but how...?
With structure!

business
cares until
here



Sounds nice,
but how...?
With structure!



business
cares until
here

automate here

Feature Set

Feature / Story

Scenario / Example

Step

Story: Returns go to stock

In order to keep track of stock

As a store owner

I want to add items back to stock when they're returned

Scenario 1: Refunded items should be returned to stock

Given a customer previously bought a black sweater from me

And I currently have three black sweaters left in stock

When he returns the sweater for a refund

Then I should have four black sweaters in stock

Scenario 2: Replaced items should be returned to stock

Given that a customer buys a blue garment

And I have two blue garments in stock

And three black garments in stock.

When he returns the garment for a replacement in black,

Then I should have three blue garments in stock

And two black garments in stock

Feature Set

Feature / Story

Scenario / Example

Step

Story: <short title>

In order <business value>

As a <role>

I want <some feature>

Feature Set

Feature / Story

Scenario / Example

Step

Scenario: <one sentence describing a specific example>

Feature Set

Feature / Story

Scenario / Example

Step

Feature Set

Feature / Story

Scenario / Example

Step



Given <context>
When <action>
Then <assertions>

Look, Ma!

No tools!



Feature Set

Feature/Story

Scenario/Example

Step

Package ←

Feature Set

Feature / Story

Scenario / Example

Step

Package ←

Feature Set

Class ←

Feature / Story

Scenario / Example

Step

Package ←

Feature Set

Class ←

Feature / Story

Method ←

Scenario / Example

Step

Package ← Feature Set

Class ← Feature/Story

Method ← Scenario/Example

Method/class ← Step

for real?

for real?



github.com/
rtens/lacarte

→ spec folder

```
--
24 class CreateUserTest extends ComponentTest {
25
26   public function setUp() {
27     parent::setUp();
28     $this->given->theGroup('test');
29     $this->given->theNextGeneratedKeyIs('myKey');
30   }
31
32   function testSuccess() {
33     $this->given->theName('Marina');
34     $this->given->theEmail('M@gnz.es');
35
36     $this->when->iCreateANewUserForTheGroup();
37
38     $this->then->theUserShouldBeCreated();
39     $this->then->thereShouldBeAUser('Marina', 'm@gnz.es');
40     $this->then->theUserShouldHaveAKey();
41   }
42
43   function testEmptyName() {
44     $this->given->theEmail('some@mail.com');
45
46     $this->when->iTryToCreateANewUserForTheGroup();
47
48     $this->then->anExceptionShouldBeThrownContaining('name');
49   }
50 }
```

mooooore ...

the inventor



<http://dannorth.net/introducing-bdd/>

<http://specificationbyexample.com/>

the book



<http://agilecoach.typepad.com/agile-coaching/2012/03/bdd-in-a-nutshell.html>

short and nice drawings



rtens.org