



Behaviour Driven Development

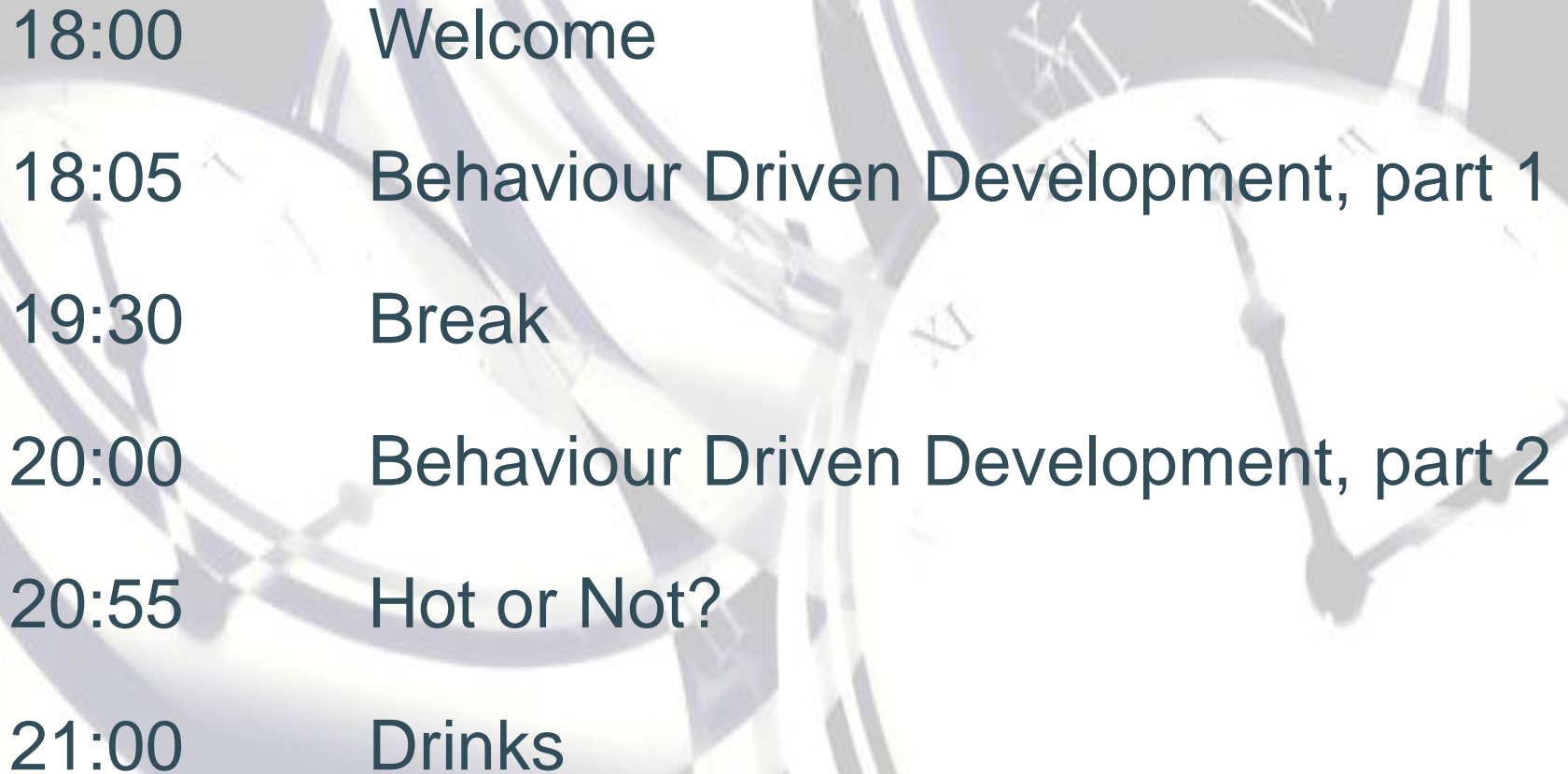
Time for a healthy talk

with Matt Wynne



Hans Odenthal

September 22nd, 2014



18:00	Welcome
18:05	Behaviour Driven Development, part 1
19:30	Break
20:00	Behaviour Driven Development, part 2
20:55	Hot or Not?
21:00	Drinks

Behaviour Driven Development



Is it about discipline?



Are fast results not healthy?

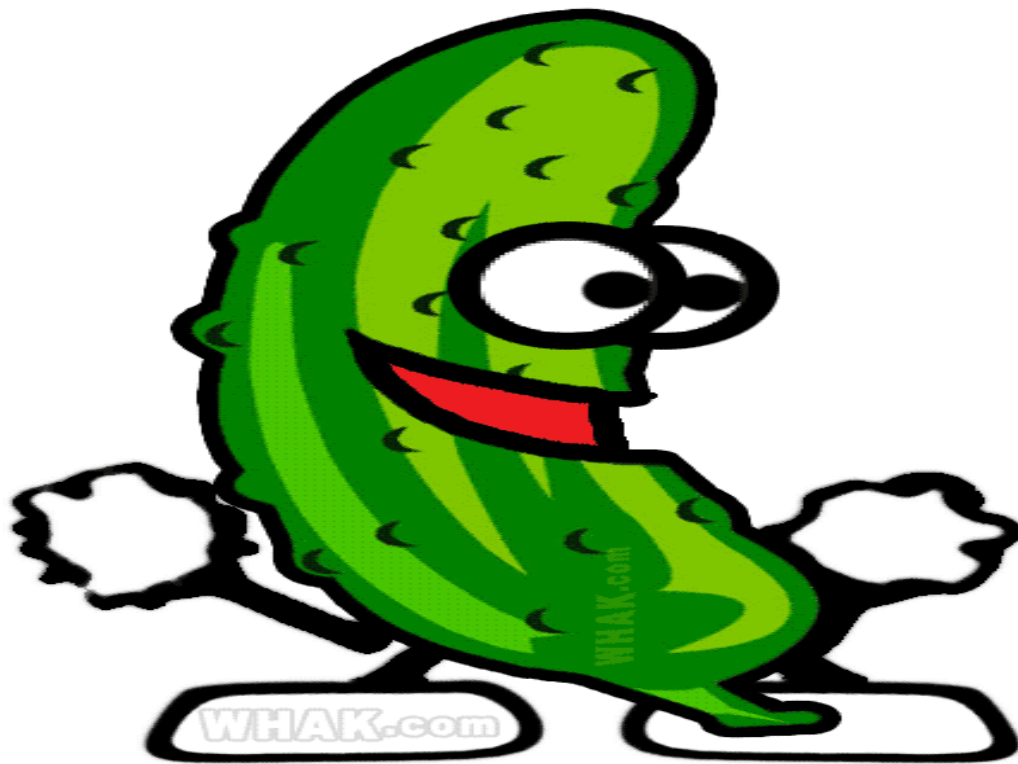


Does green always mean good?

Matt will give
all the answers



Matt, the floor is yours ...



Behaviour-Driven Development

Matt Wynne
Sioux Embedded Systems, Eindhoven
22 September 2014

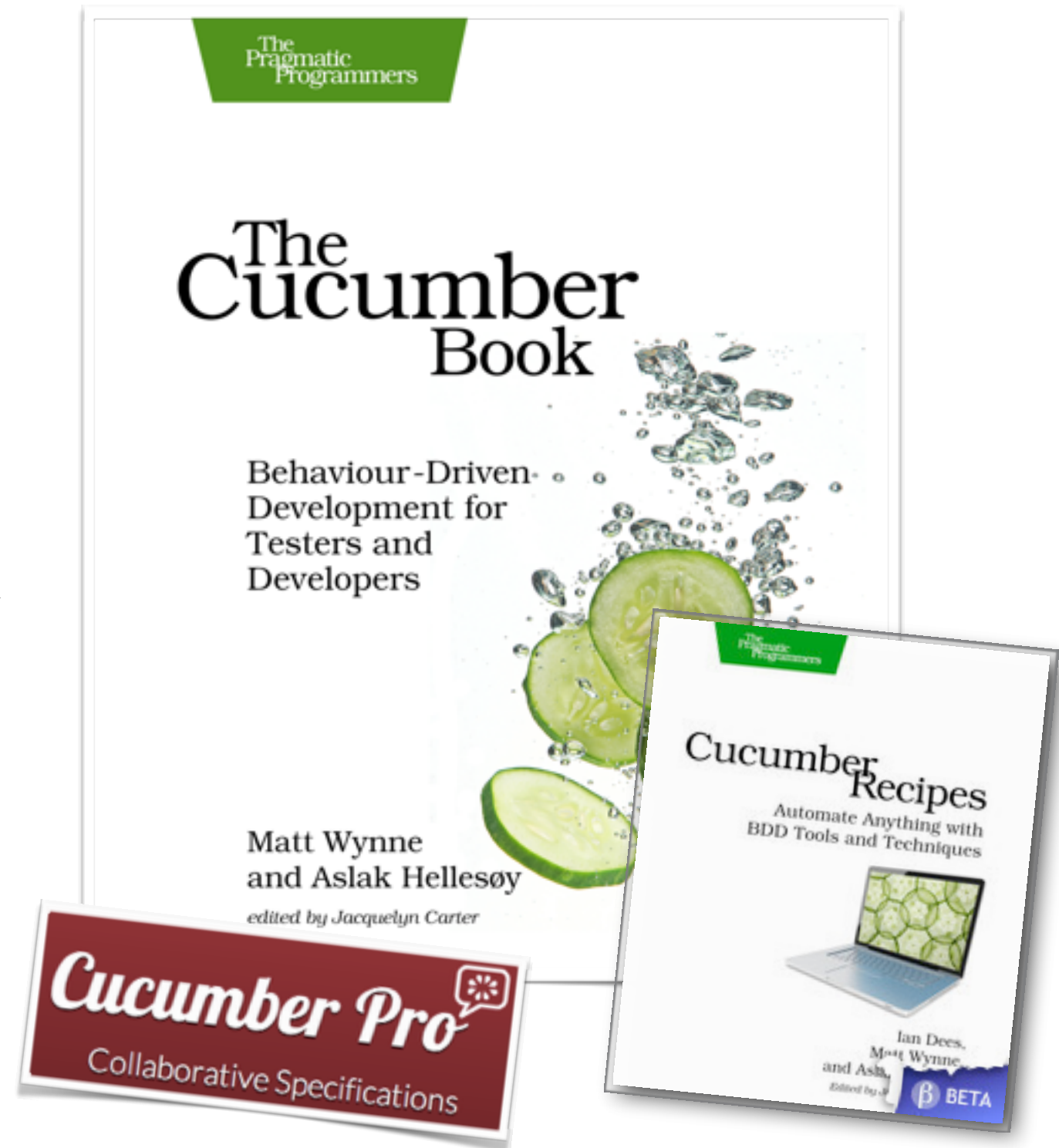
@mattwynne
matt@cucumber.pro

What is
BDD?

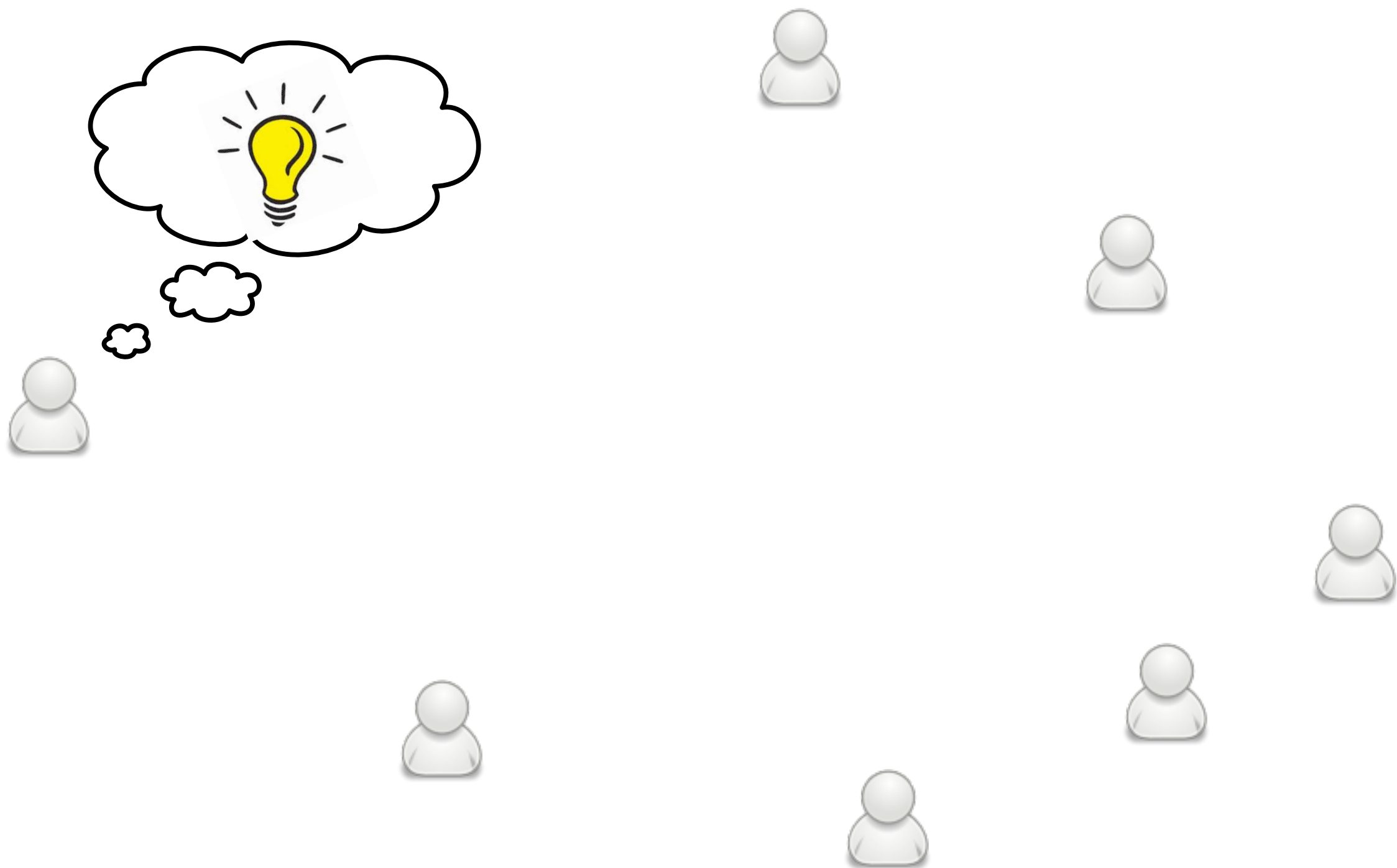
...and why should I care?

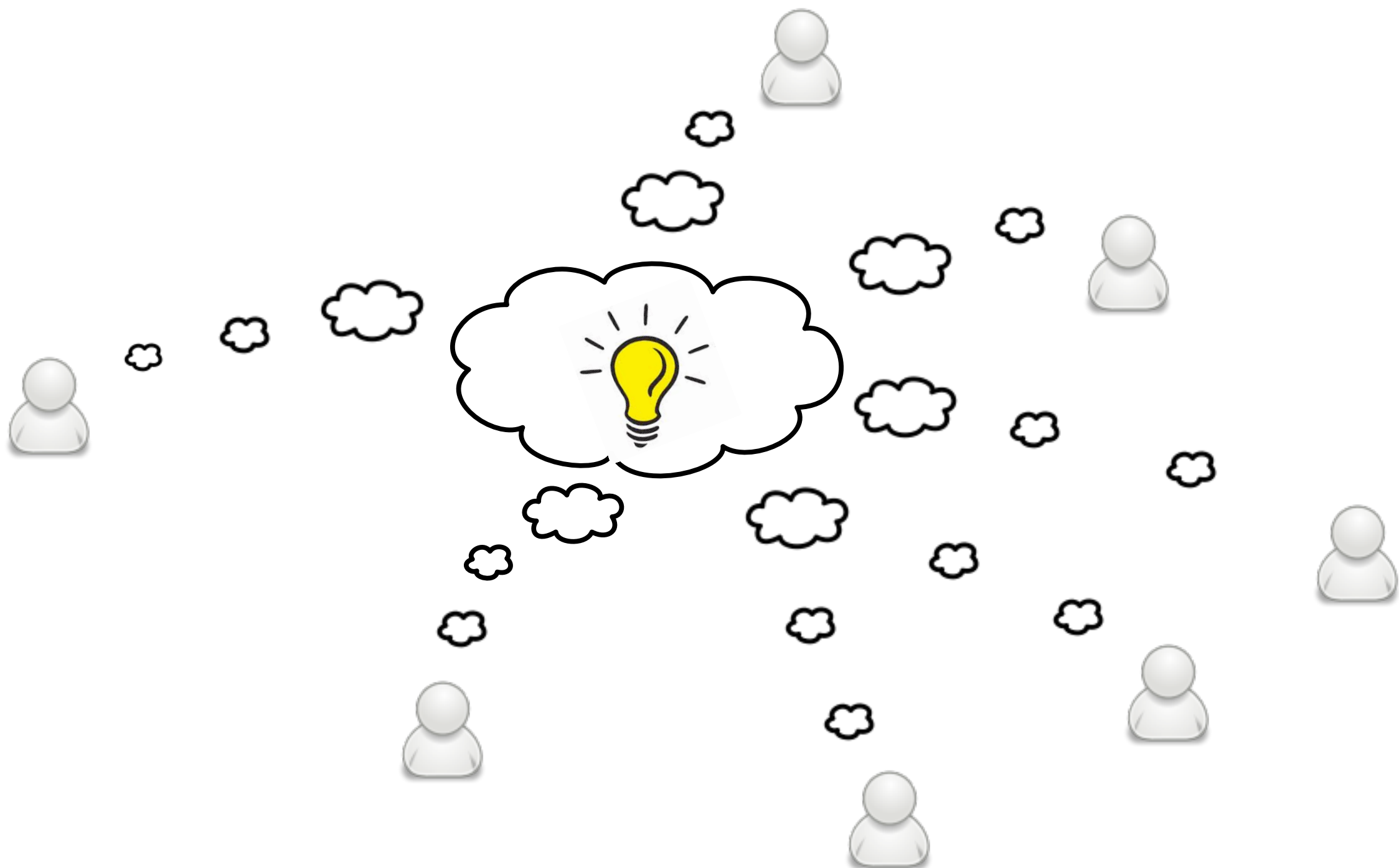
About Matt

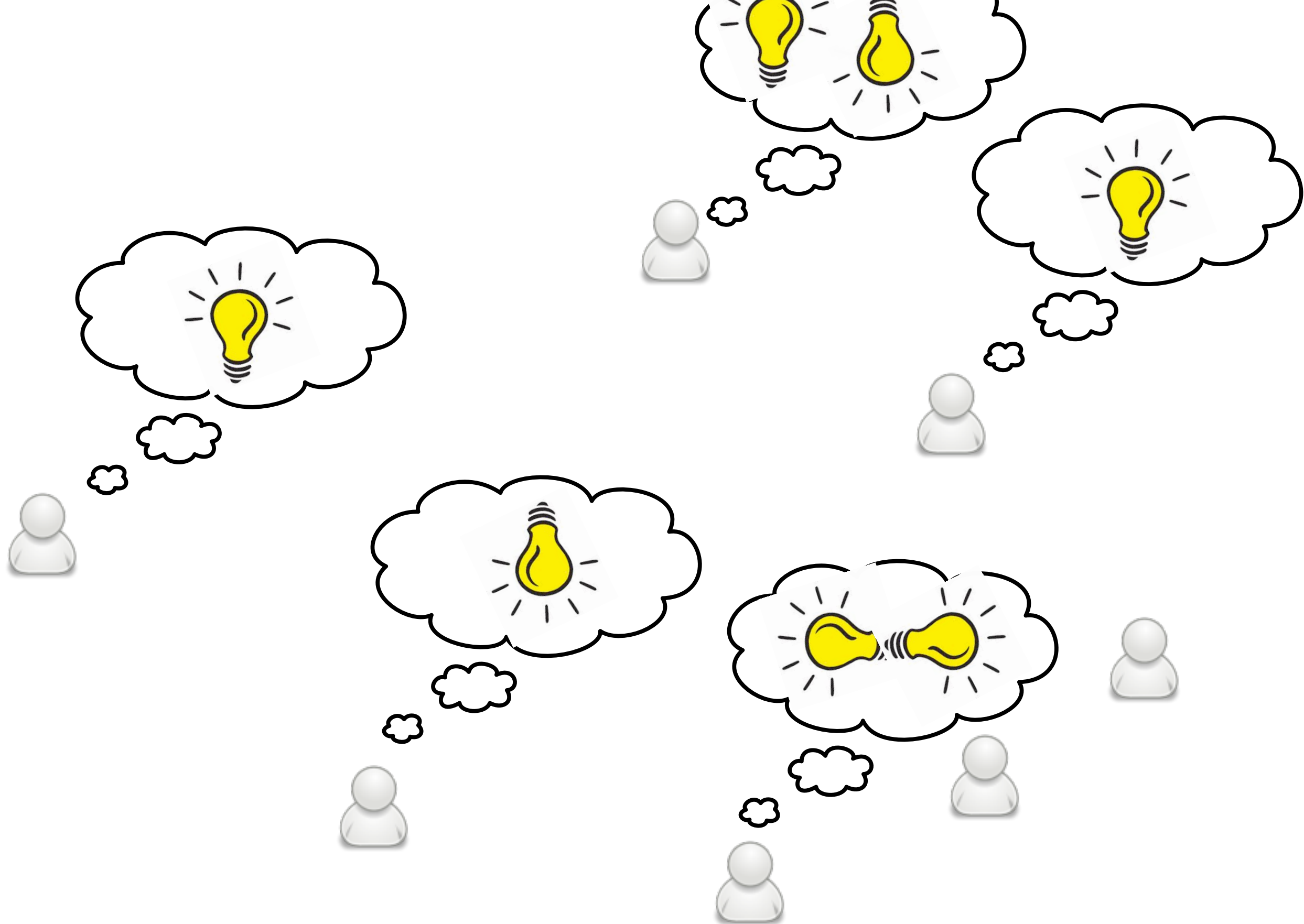
- Programmer
- Agile / lean advocate & coach
- Author of *The Cucumber Book*
- Founder of Cucumber Ltd









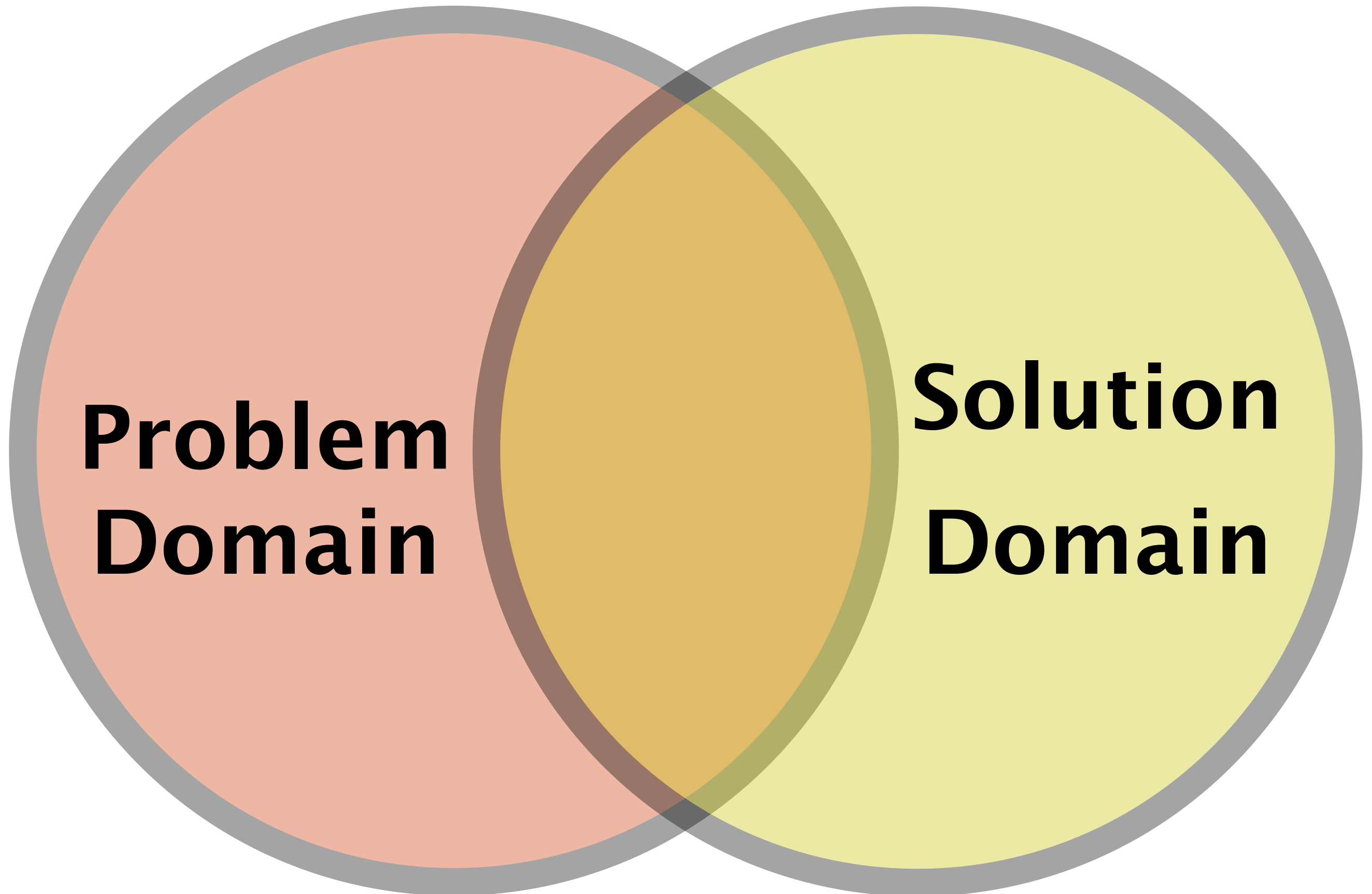




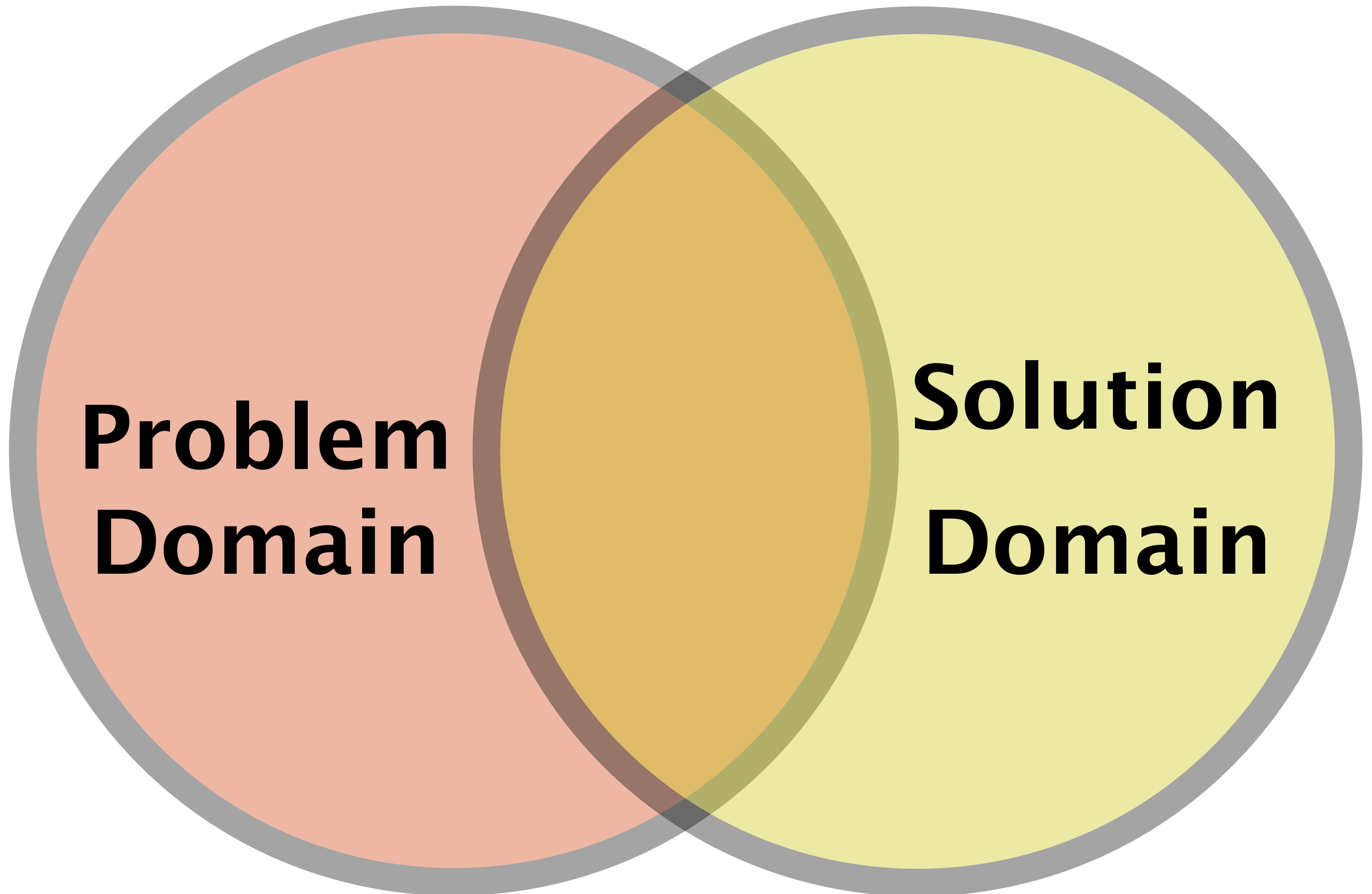
Stakeholders,
Customers,
Users

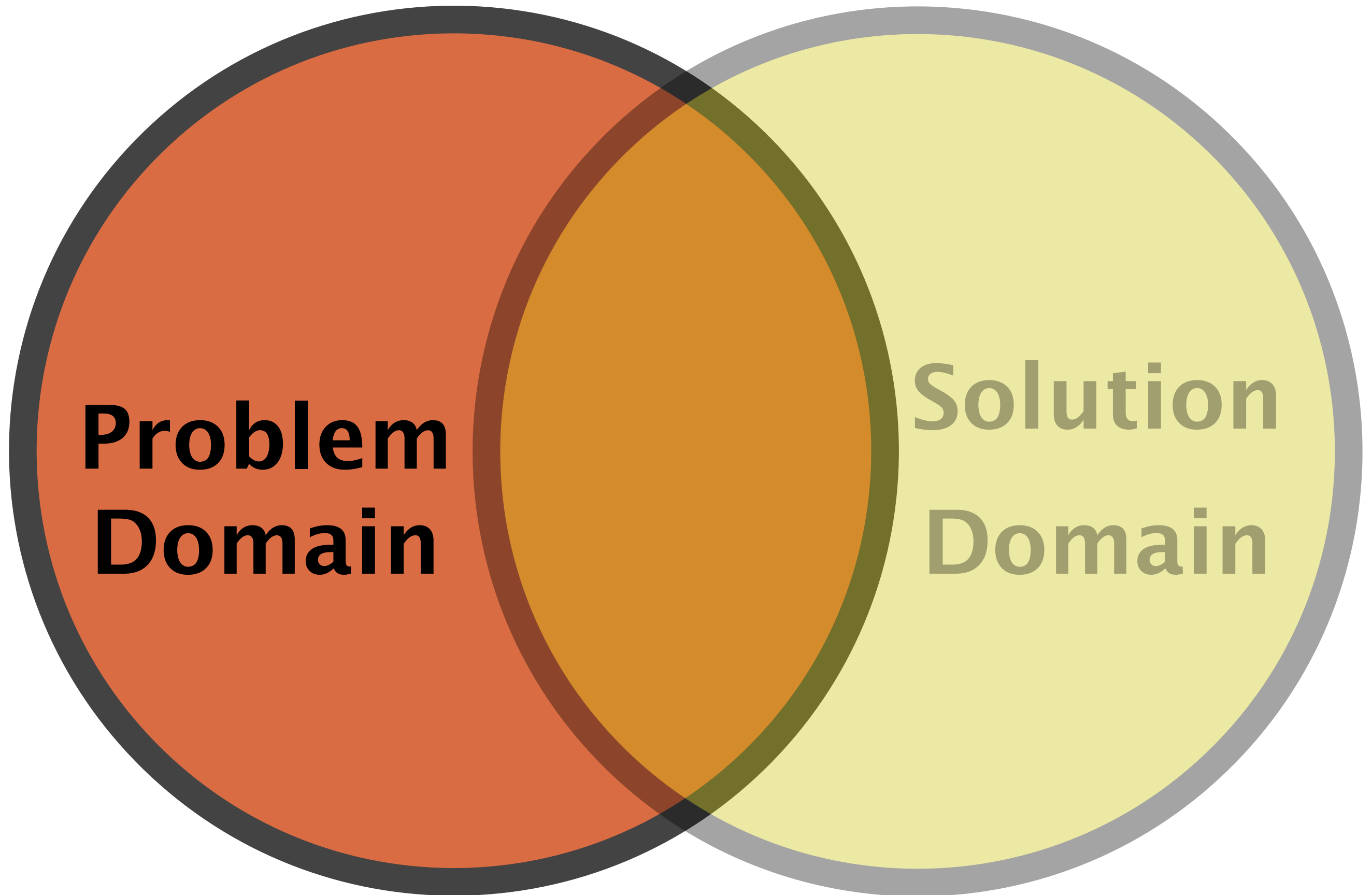


Programmers,
Testers



Ubiquitous Language





**Problem
Domain**

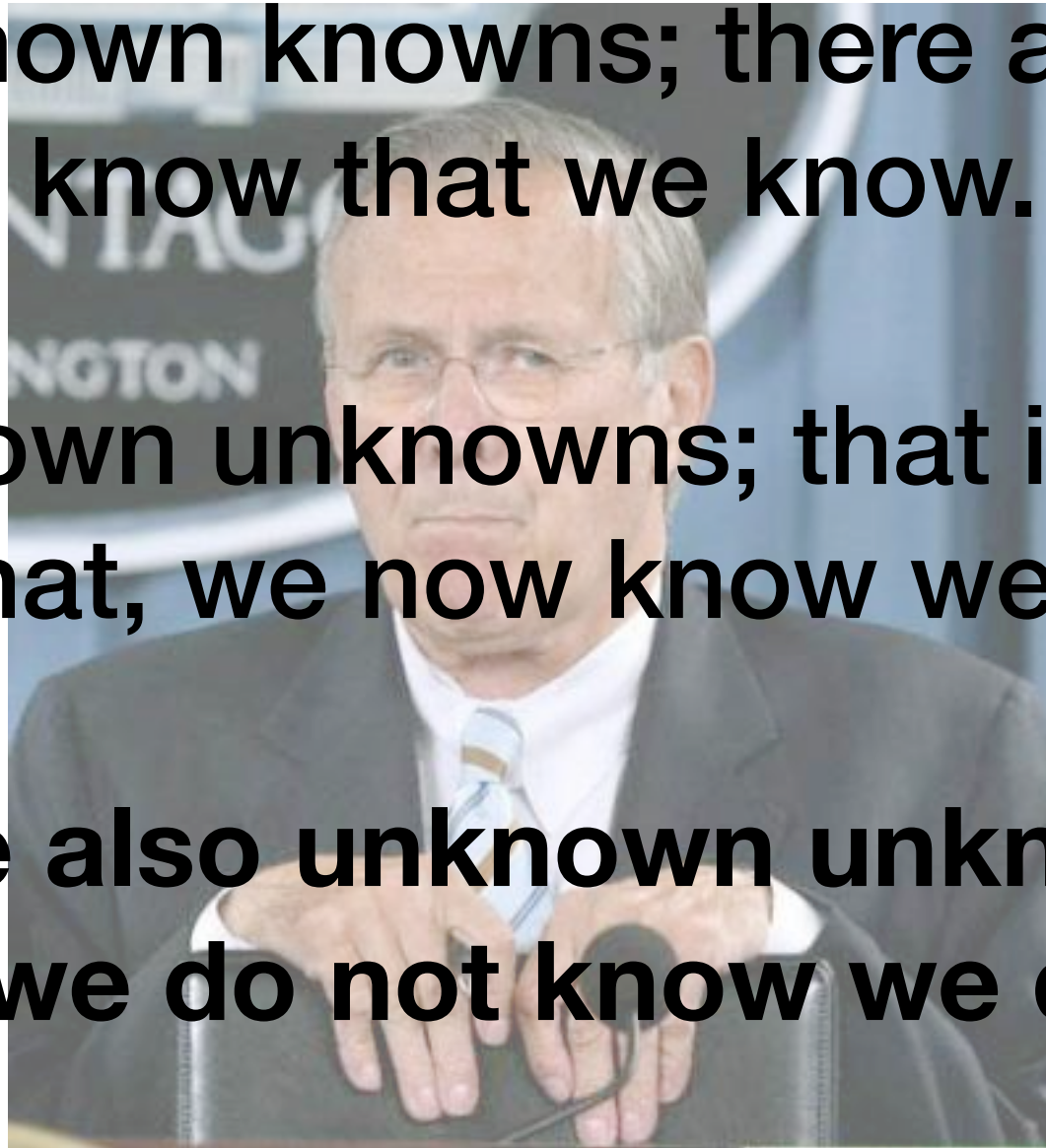
**Solution
Domain**

Deliberate discovery

There are known knowns; there are things we know that we know.

There are known unknowns; that is to say there are things that, we now know we don't know.

But there are also unknown unknowns – there are things we do not know we don't know.



ASSUME

YOU'RE

IGNORANT

Specification by
example

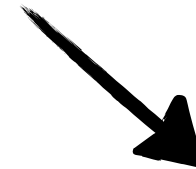
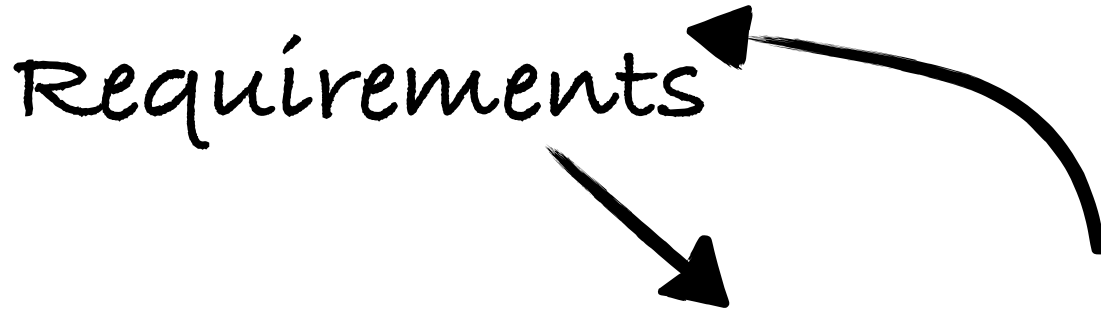
Ideas

Requirements

Examples

Automated Tests

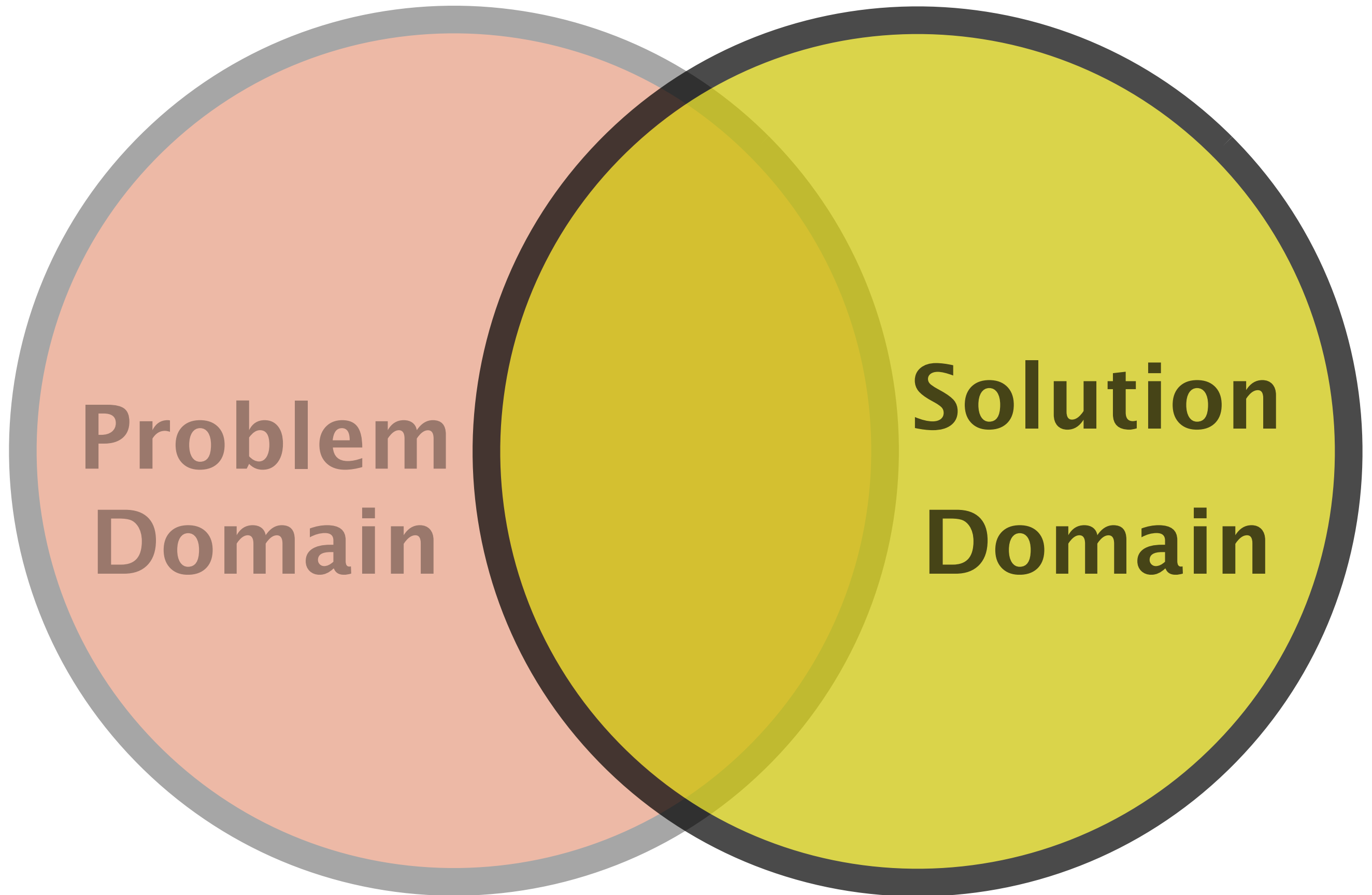
Code



CONVERSATIONS

The Three Amigos





**Problem
Domain**

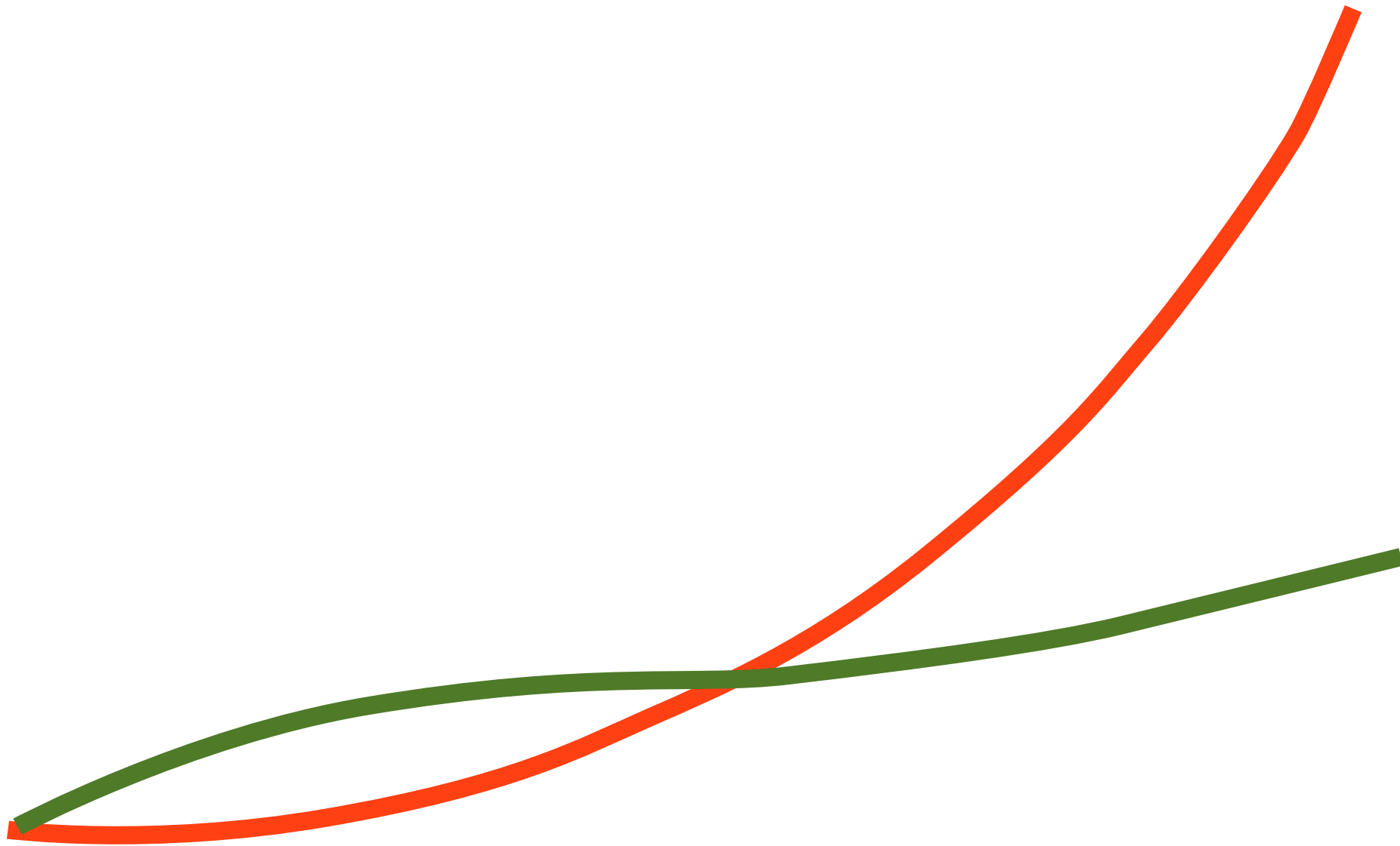
**Solution
Domain**

Sustainable delivery

Cost of a
change



Time







You can't stay agile
without clean code

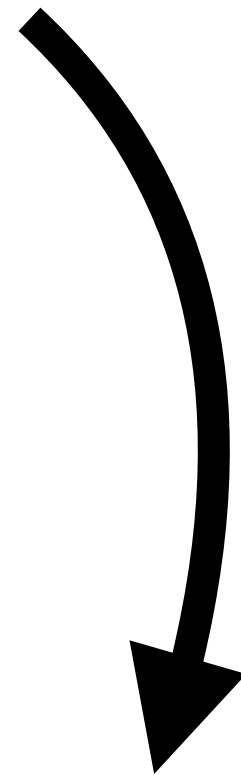
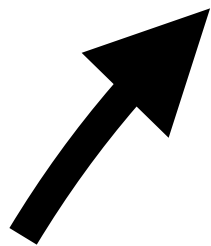
You can't have clean code
without refactoring

You can't refactor without
good automated tests

Write a failing
test

Clean
up!

Make it pass



Living documentation

Living Documentation
Quick Start Guide
Admin
Api
Billing
Client gem
Dashboard

Guest access

- **Enable guest access**
- › Guest access via URL
- › Sharing Links

Home Page
Projects
Publishers
Search
Users
Visitors

Enable guest access



It should be possible to enable (and disable) guest access to a project via the [project's settings page](#).

Background

Given there is a private project
And I am signed in
And I am a collaborator on the project

Scenarios

- ▼ Enable
- ▼ Disable
- ▼ Don't see the option for a public project

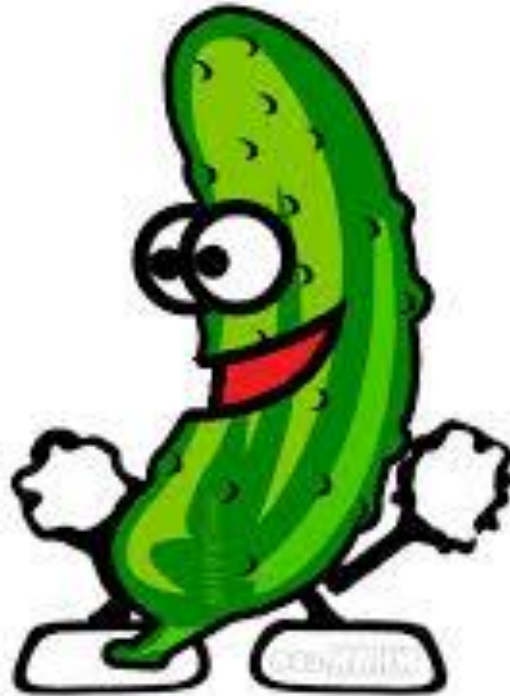
◦ Enable

When I visit the project's settings page
Then I should see that guest access is not enabled
When I enable guest access via the button on the page
Then guest access should be enabled for the project
And I should see a link to gain guest access to the project

So what is BDD?

- **Using examples to drive conversations**
- **Using conversations to grow the team's domain knowledge**
- **Using automated tests for the examples, to keep total cost of ownership under control**

The floor is yours ...



MATT WYNNE, SOIUX EINDHOVEN 2014

LIFE AFTER BDD



AGILE



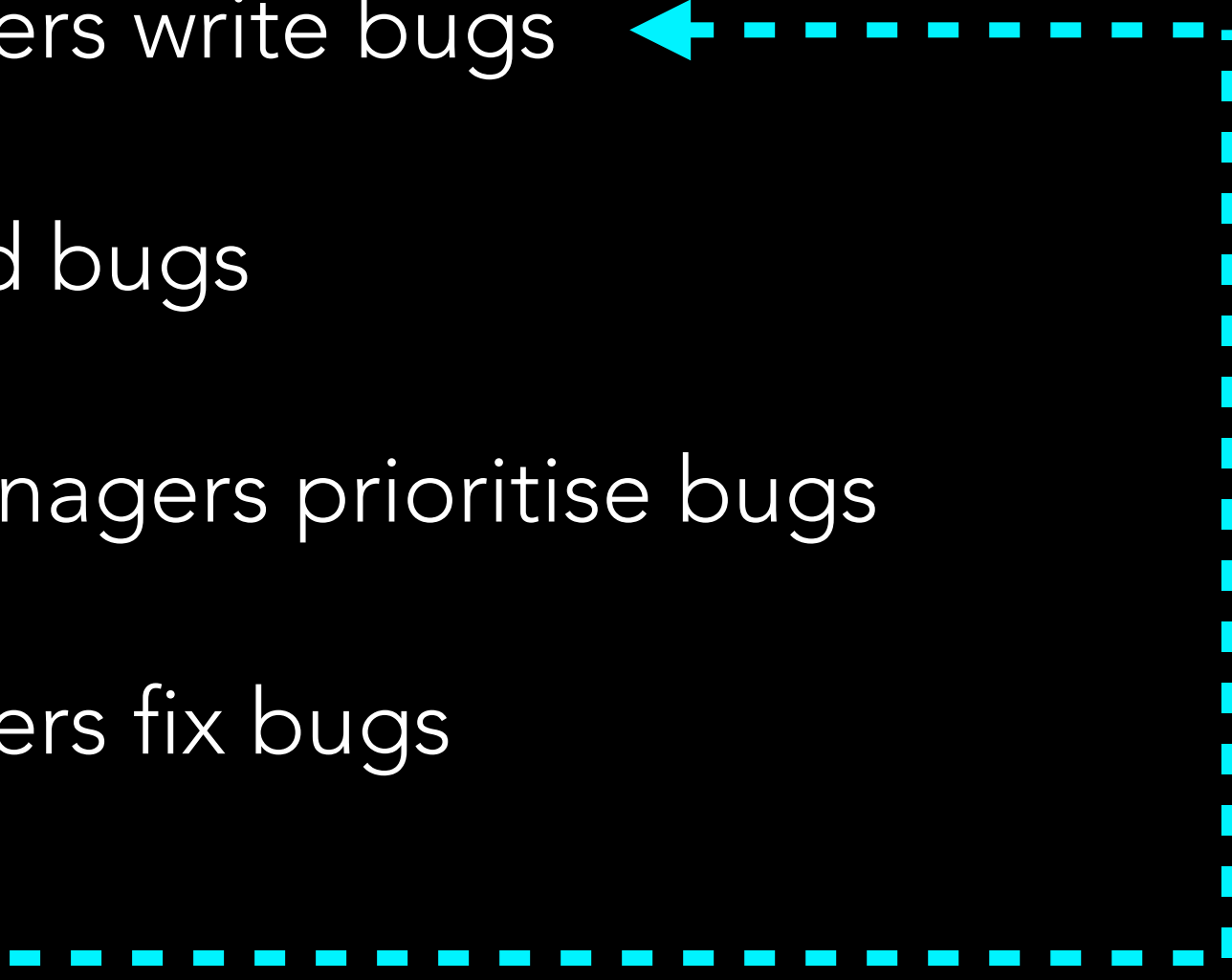
SCRUM



STAGE 1:

BURNED TOAST

STAGE 1: BURNED TOAST

1. Programmers write bugs
 2. Testers find bugs
 3. Project managers prioritise bugs
 4. Programmers fix bugs
 5. GOTO 1
- 
- ```
graph TD; 1[1. Programmers write bugs] --> 2[2. Testers find bugs]; 2 --> 3[3. Project managers prioritise bugs]; 3 --> 4[4. Programmers fix bugs]; 4 --> 5[5. GOTO 1]; 5 --> 1;
```

"LET'S MAKE TOAST THE AMERICAN WAY:

I'LL BURN IT,  
YOU SCRAPE IT"

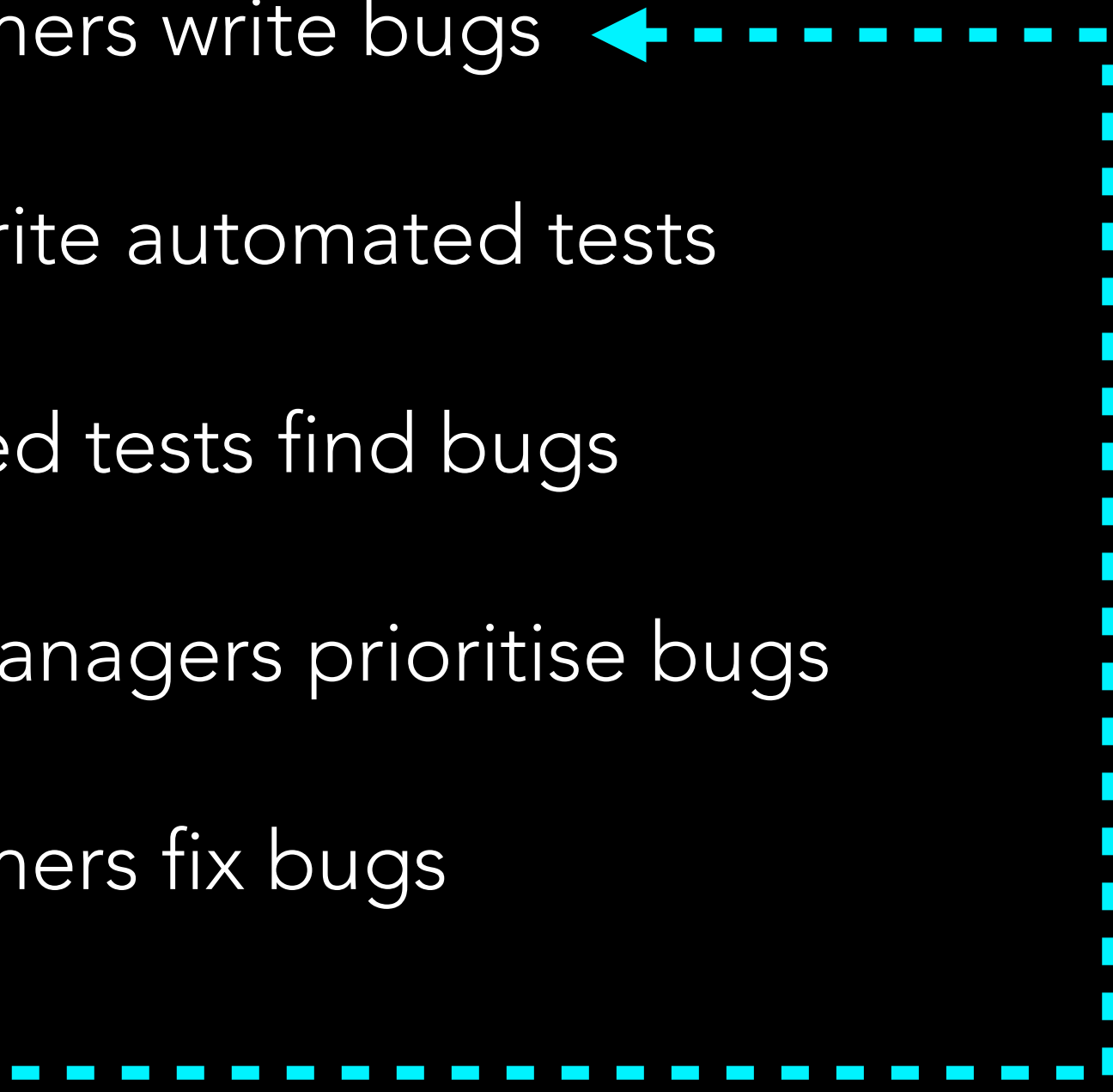
-DEMMING



STAGE 2:

AUTO-BURNED TOAST

# STAGE 2: AUTO-BURNED TOAST

1. Programmers write bugs
  2. Testers write automated tests
  3. Automated tests find bugs
  4. Project managers prioritise bugs
  5. Programmers fix bugs
  6. GOTO 1
- 
- ```
graph TD; 1[1. Programmers write bugs] --> 2[2. Testers write automated tests]; 2 --> 3[3. Automated tests find bugs]; 3 --> 4[4. Project managers prioritise bugs]; 4 --> 5[5. Programmers fix bugs]; 5 --> 6[6. GOTO 1]; 6 --> 1;
```


THAT IS

NOT

BDD

EVEN IF
YOU'RE
USING
CUCUMBER

BUILDING SOFTWARE BACKWARDS



BEHAVIOUR

DRIVEN

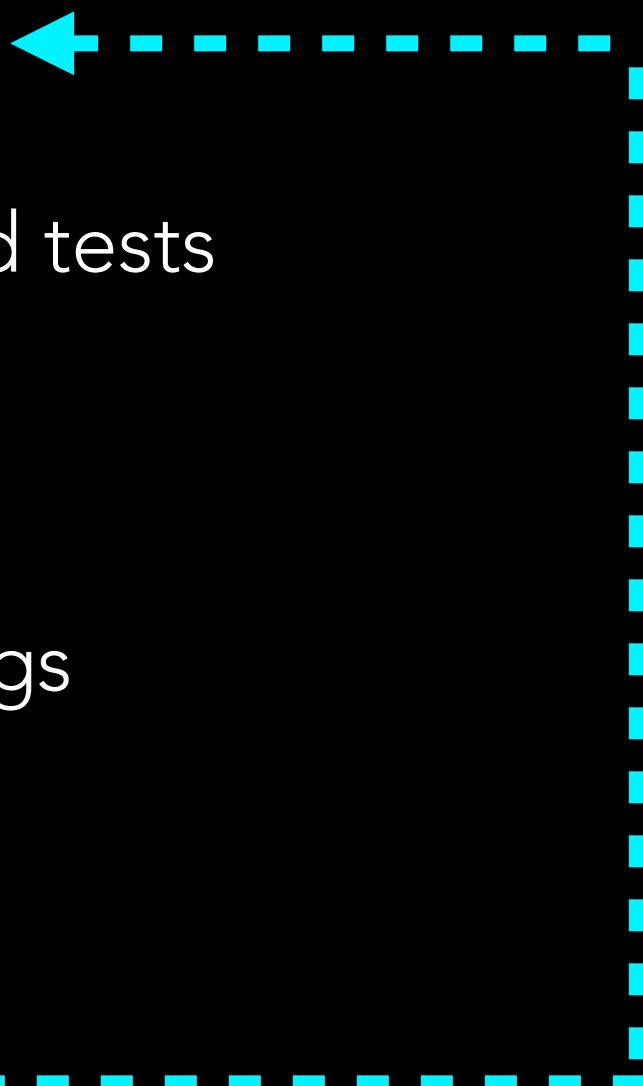
DEVELOPMENT



STAGE 3:

THREE AMIGOS

STAGE 3: THREE AMIGOS

1. Programmers, Testers and BAs define behaviour together
 2. Programmers write (less) bugs ←
 3. Testers write (more) automated tests
 4. Automated tests find bugs
 5. Project managers prioritise bugs
 6. Programmers fix bugs
 7. GOTO 2
- 
- A dashed red line starts at the end of step 2, extends horizontally to the right, then turns vertically downwards, and finally turns horizontally to the left, ending with an arrowhead pointing to the left side of step 2. This indicates a feedback loop from the final step back to step 2.

story

question

question

Rule

Rule

Rule

example

example

example

STAGE 3: THREE AMIGOS

1. Programmers, Testers and BAs define behaviour together
2. Programmers write (less) bugs
3. Testers write (more) automated tests
4. Automated tests find bugs
5. Project managers prioritise bugs
6. Programmers fix bugs
7. GOTO 2

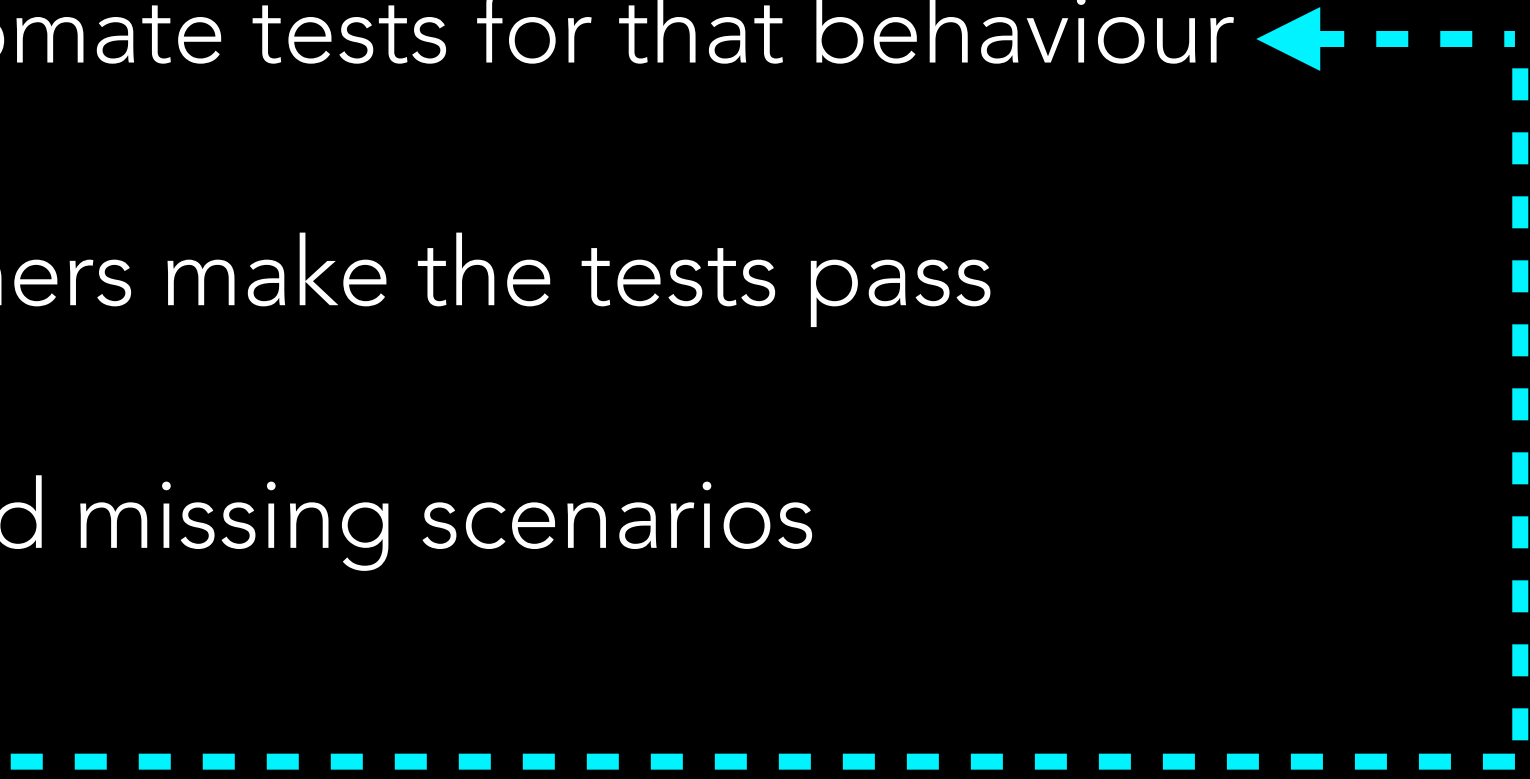


WHY DO
TESTERS HAVE
TO AUTOMATE
THE TESTS?

STAGE 4:

TEST-FIRST

STAGE 4: TEST-FIRST

1. Programmers, Testers and BAs define behaviour together
 2. Team automate tests for that behaviour
 3. Programmers make the tests pass
 4. Testers find missing scenarios
 5. GOTO 2
- 
- ```
graph TD; 1[1. Programmers, Testers and BAs define behaviour together] --> 2[2. Team automate tests for that behaviour]; 2 --> 3[3. Programmers make the tests pass]; 3 --> 4[4. Testers find missing scenarios]; 4 --> 5[5. GOTO 2]; 5 --> 2;
```



BDD



STAGE 5:

DISILLUSIONMENT

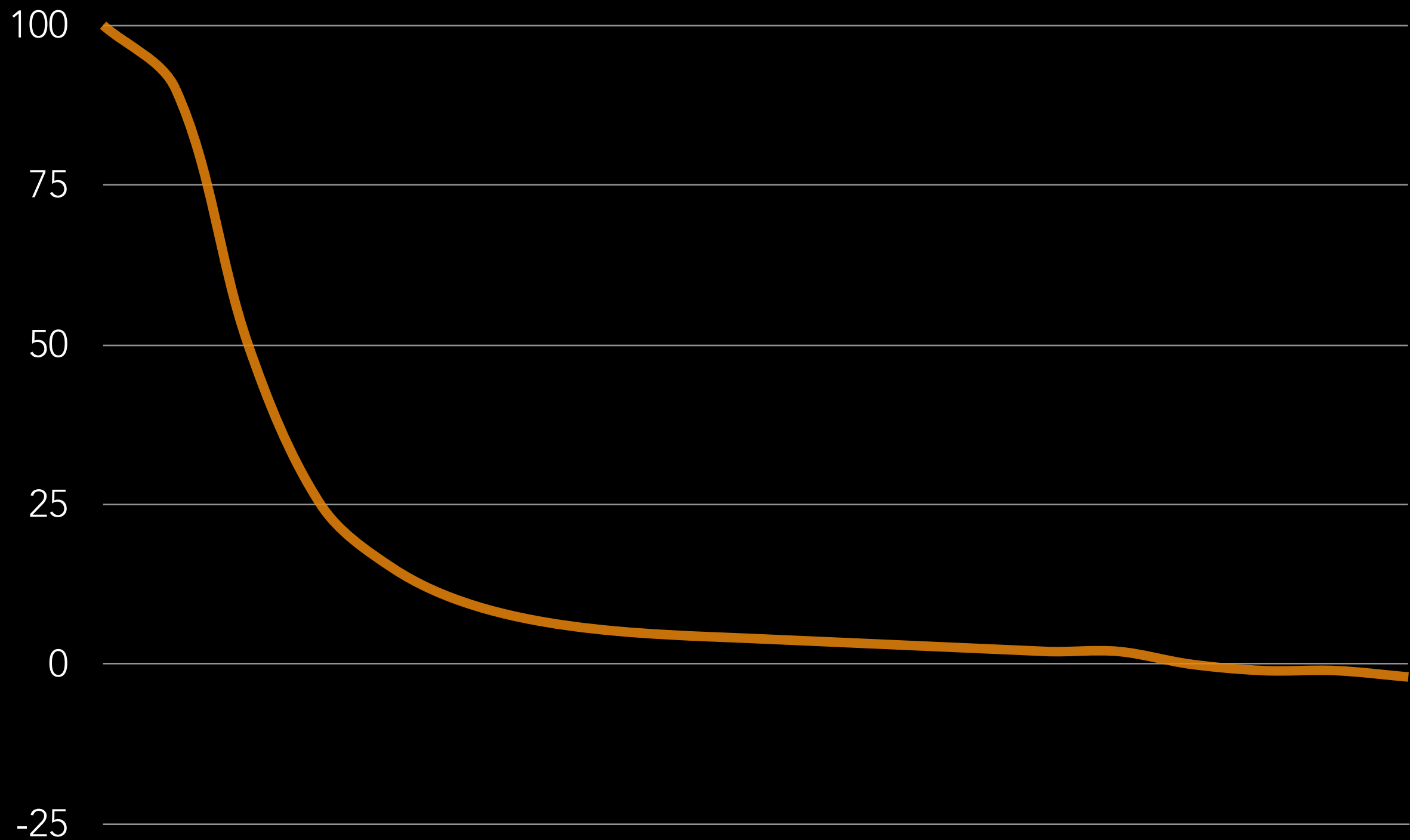
# STAGE 5: DISILLUSIONMENT

- Lots of scenarios
- Build takes ages
- Build normally broken
- Some scenarios flicker
- Poor / mixed readability
- I hate Cucumber





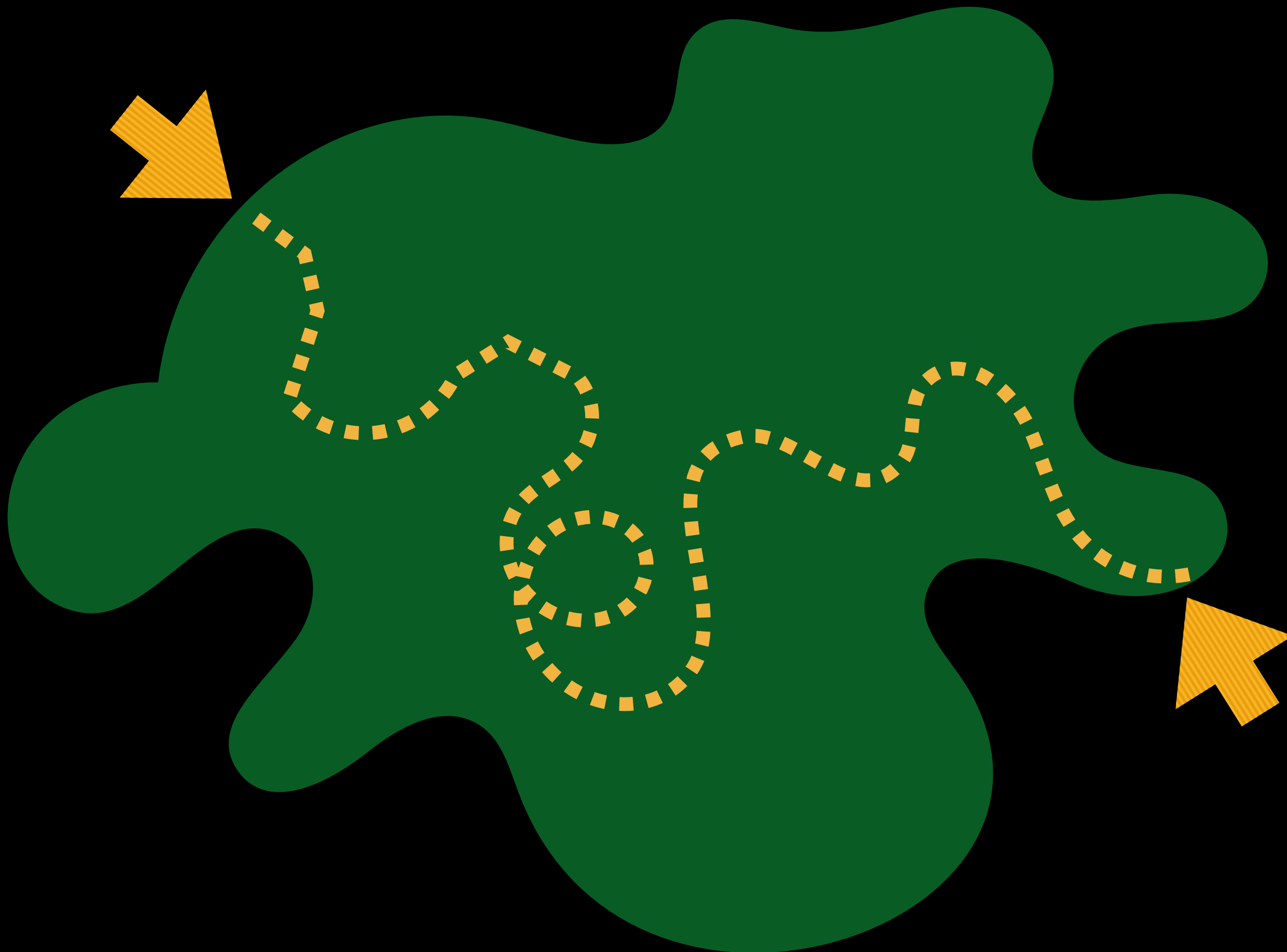
# VALUE OF A TEST OVER TIME

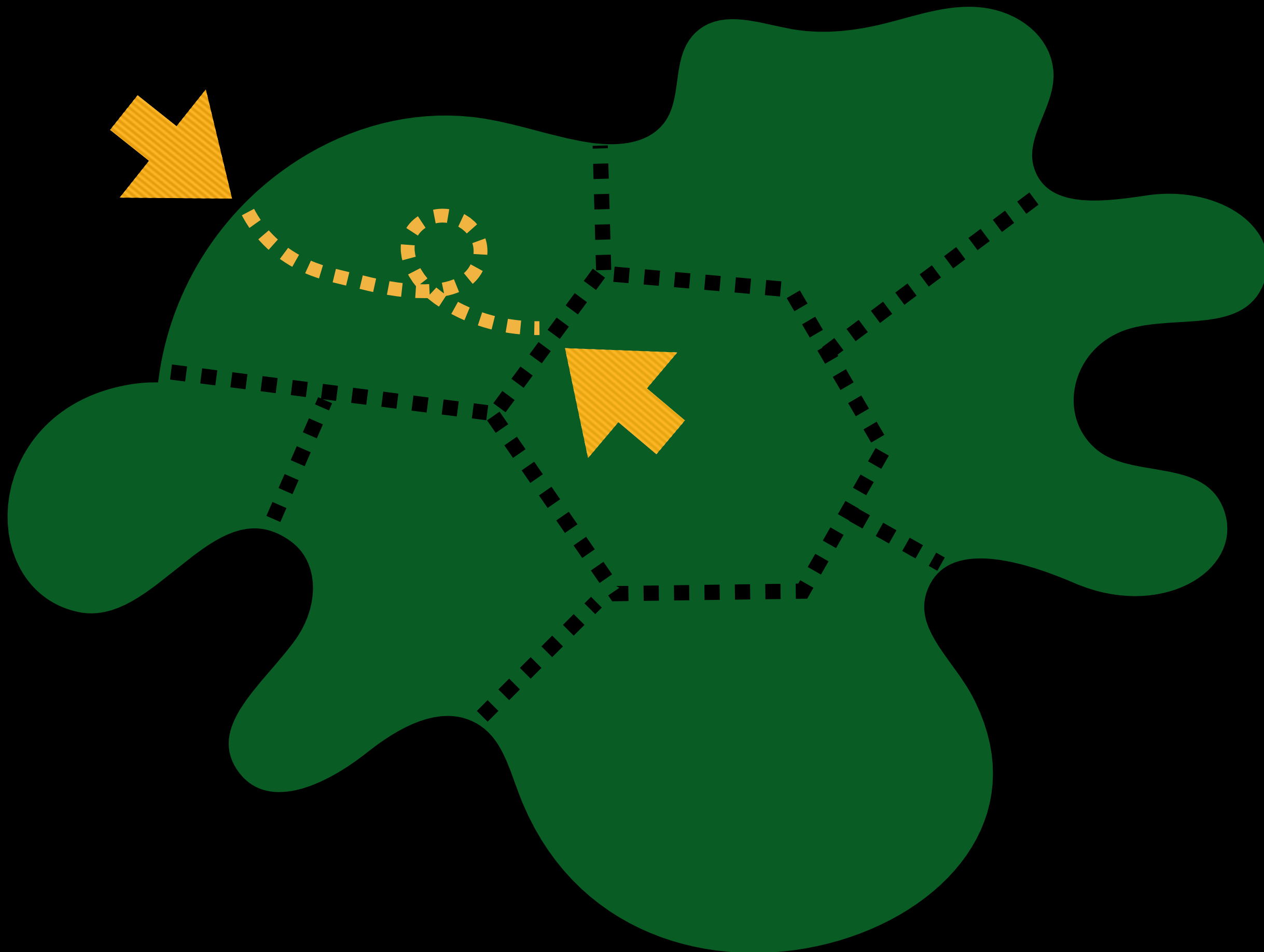


WHEN'S THE  
LAST TIME  
YOU PUSHED A  
TEST DOWN?

*(or just deleted it altogether)*

YOU STILL  
HAVE TO  
DO  
SOFTWARE  
DESIGN







STAGE 6:

TRANSCENDENCE



# STAGE 6: TRANSCENDENCE

- Problem domain is well understood by the team
- Solution models the problem well
- Scenarios are actually readable
- Rules over examples
- Clear architectural boundaries
- Fewer end-to-end tests

THE END



# Hot-or-Not?





# ‘Komkommertijd’?



Oct 22 littleBits Workshop  
*Hot-or-Not, The Next Generation*



Nov 10 Markus Völter & Tijs van der Storm  
*MDSD: What's the right way to go?*



*More information: [www.siox.eu/hotornot](http://www.siox.eu/hotornot)*

September 23 - 24, 2014

Behaviour Driven Development

*Matt Wynne*

November 11 - 12, 2014

Language Engineering with MPS

*Markus Völter*

Thank you







SOURCE OF YOUR TECHNOLOGY



[www.siox.eu](http://www.siox.eu)



+31(0)40 2677100



[hans.odenthal@siox.eu](mailto:hans.odenthal@siox.eu)