Behavior Driven Development

&

Specification ~~by~~ Examples

With

Arnauld Loyer - @aloyer

Authentic Developer

since 2000!

Arnauld Loyer

@aloyer

Software Craftsman

BDD fan boy

arolla

It's not about tools!

It's about Communication and Behavior!

It's not about tools!

It's not about testing!

It's about Exploring the Unknown

It's not about testing!

It's about Exploring the Unknown

It's about Driven Development!

It's not about testing!


It's about Sharing and Understanding
It's not about Knowledge Silo!

Testers

Developers

Business

Chaos Report

http://blog.standishgroup.com/

Chaos Report

http://blog.standishgroup.com/

Project Challenged Factors	
1. **Lack of User Input**	**12.8%**
2. **Incomplete Requirements & Specifications**	**12.3%**
3. Changing Requirements & Specifications	11.8%
4. Lack of Executive Support	7.5%
5. Technology Incompetence	7.0%
6. Lack of Resources	6.4%
7. Unrealistic Expectations	5.9%
8. Unclear Objectives	5.3%
9. Unrealistic Time Frames	4.3%
10. New Technology	3.7%

Project Impaired Factors	% of
1. **Incomplete Requirements**	**13.1%**
2. **Lack of User Involvement**	**12.4%**
3. Lack of Resources	10.6%
4. Unrealistic Expectations	9.9%
5. Lack of Executive Support	9.3%
6. Changing Requirements & Specifications	8.7%
7. Lack of Planning	8.1%
8. Didn't Need It Any Longer	7.5%
9. Lack of IT Management	6.2%
10. Technology Illiteracy	4.3%

Chaos Report

http://blog.standishgroup.com/

The Bull Survey (1998)

All the managers interviewed had previously taken the lead in integrating large systems within organizations in the Times Top 100.

Project Evaluation Criteria

The main IT project failure criteria identified by the IT and project managers were:

* missed deadlines (75%)
* exceeded budget (55%)
* **poor communications** (40%)
* **inability to meet project requirements** (37%).

http://www.it-cortex.com/Stat_Failure_Cause.htm

The Bull Survey (1998)

had previously taken the lead in integrating large systems

2005 Survey: The Top Challenges Facing Business Analysts

Business Improvement Architect's research survey of Business Analysts attending Project World/ Business Analyst World 2005 in Toronto, Canada, identified that 'Lack of Clarity in the Scope of the Business Functions' and 'Business Requirements Not Well-Defined" are the top two challenges facing their organizations in managing business requirements.

-- http://www.bia.ca/articles/TheTopChallengesFacingBusinessAnalysts.htm

* exceeded budget (
* **poor communications** (40%)
* **inability to meet project requirements** (37%).

http://www.it-cortex.com/Stat_Failure_Cause.htm

Why IT Projects Fail

1. Lack of Governance
2. Internal Politics
3. Poor communication between business and IT
4. Unclear expectations

http://www.itbusinessedge.com/slideshows/show.aspx?c=81818&slide=4

had previously taken the lead in integrating large systems

The Top Challenges Facing Business Analysts

Improvement Architect's research survey of Business Analysts attending Project World/ Business Analyst World 2005 in Toronto, Canada, identified that 'Lack of Clarity in the Scope of the Business Functions' and 'Business Requirements Not Well-Defined" are the top two challenges facing their organizations in managing business requirements.

-- http://www.bia.ca/articles/TheTopChallengesFacingBusinessAnalysts.htm

* exceeded budget (
* **poor communications** (40%)
* **inability to meet project requirements** (37%).

http://www.it-cortex.com/Stat_Failure_Cause.htm

Why IT Projects Fail

1. Lack of Governance
2. Internal Politics
3. Poor communication between business and IT
4. Unclear expectations

Why Software Fails - We waste billions of dollars each year on entirely preventable mistakes (2005)
http://spectrum.ieee.org/computing/software/why-software-fails

Among the most common factors:

* Unrealistic or unarticulated project goals
* Inaccurate estimates of needed resources
* **Badly defined system requirements**
* Poor reporting of the project's status
* Unmanaged risks
* **Poor communication among customers, developers, and users**
* Use of immature technology
* Inability to handle the project's complexity
* Sloppy development practices
* Poor project management
* Stakeholder politics
* Commercial pressures

Seven Reasons IT Projects Fail - February 2012

1. *Poor Project Planning and Direction*
2. *Insufficient Communication*
3. *Ineffective Management*
4. *Failure to Align With Constituents and Stakeholders*

http://www.ibmsystemsmag.com/power/Systems-Management/Workload-Management/project_pitfalls/?page=2

Among the most com
* Unrealistic or unarticulated project g
* Inaccurate estimates of needed resources
* **Badly defined system requirements**
* Poor reporting of the project's status
* Unmanaged risks
* **Poor communication among customers, developers, and users**
* Use of immature technology
* Inability to handle the project's complexity
* Sloppy development practices
* Poor project management
* Stakeholder politics
* Commercial pressures

Why IT
1. Lack
2. Inter
3. Poo
4. Un

Seven Reasons IT Projects Fail - February 2012

1. *Poor Project Planning and Direction*
2. *Insufficient Communication*
3. *Ineffective Management*
4. *Failure to Align With Constituents and Stakeholders*

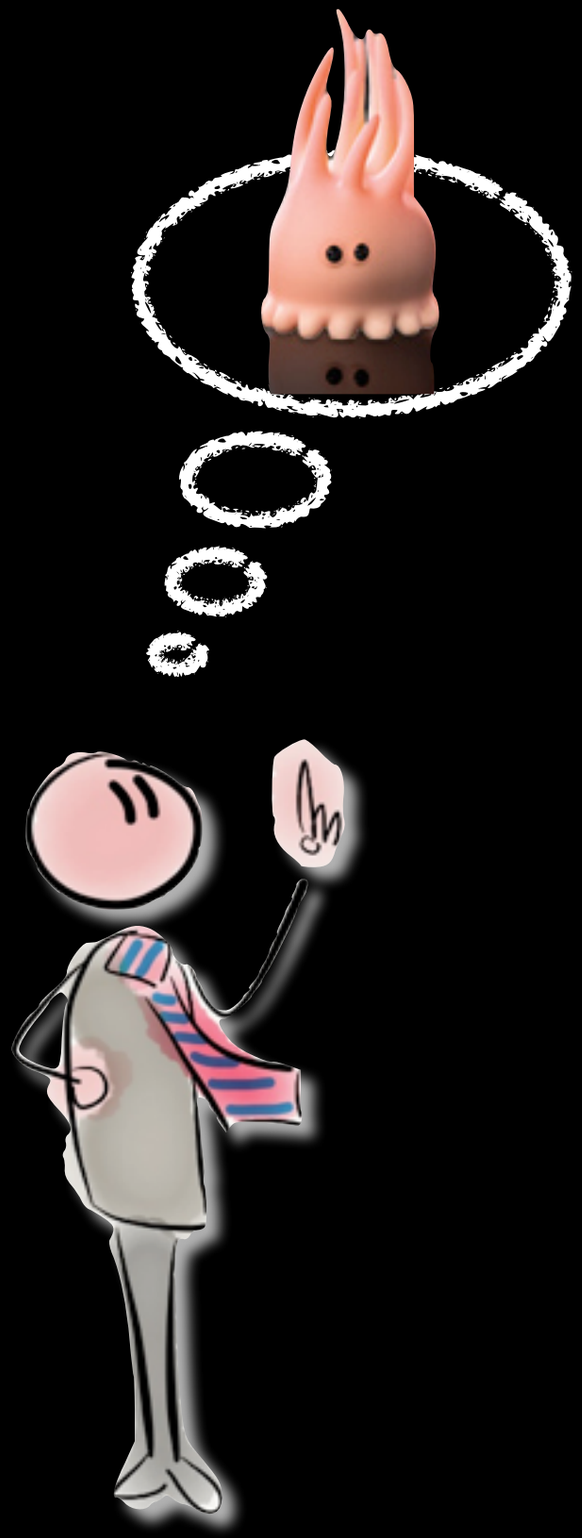
http://www.ibmsystemsmag.com/power/Systems-Management/Workload-Management/project_pitfalls/?page=2

The Most Common Factors for the failure of Software Development Project

1. Lack of Customer or User Involvement
2. Unclear goals and objectives
3. Poor Requirement Set
4. Lack of Resources
5. Failure to communicate and act as a team

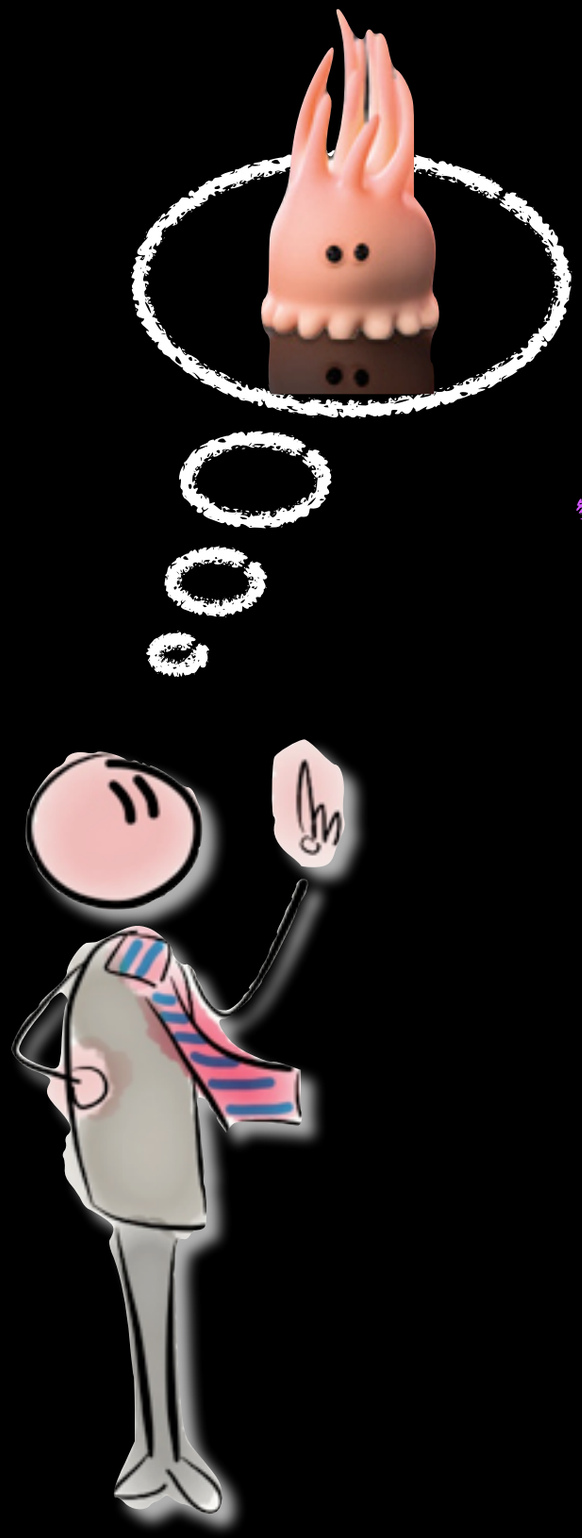
* Stakeh
* Commercial pressures

Volume 1, No. 11, January 2013 ISSN – 2278-1080
The International Journal of Computer Science & Applications
http://www.journalofcomputerscience.com/2013Issue/Jan13/V1No11Jan13P044.pdf



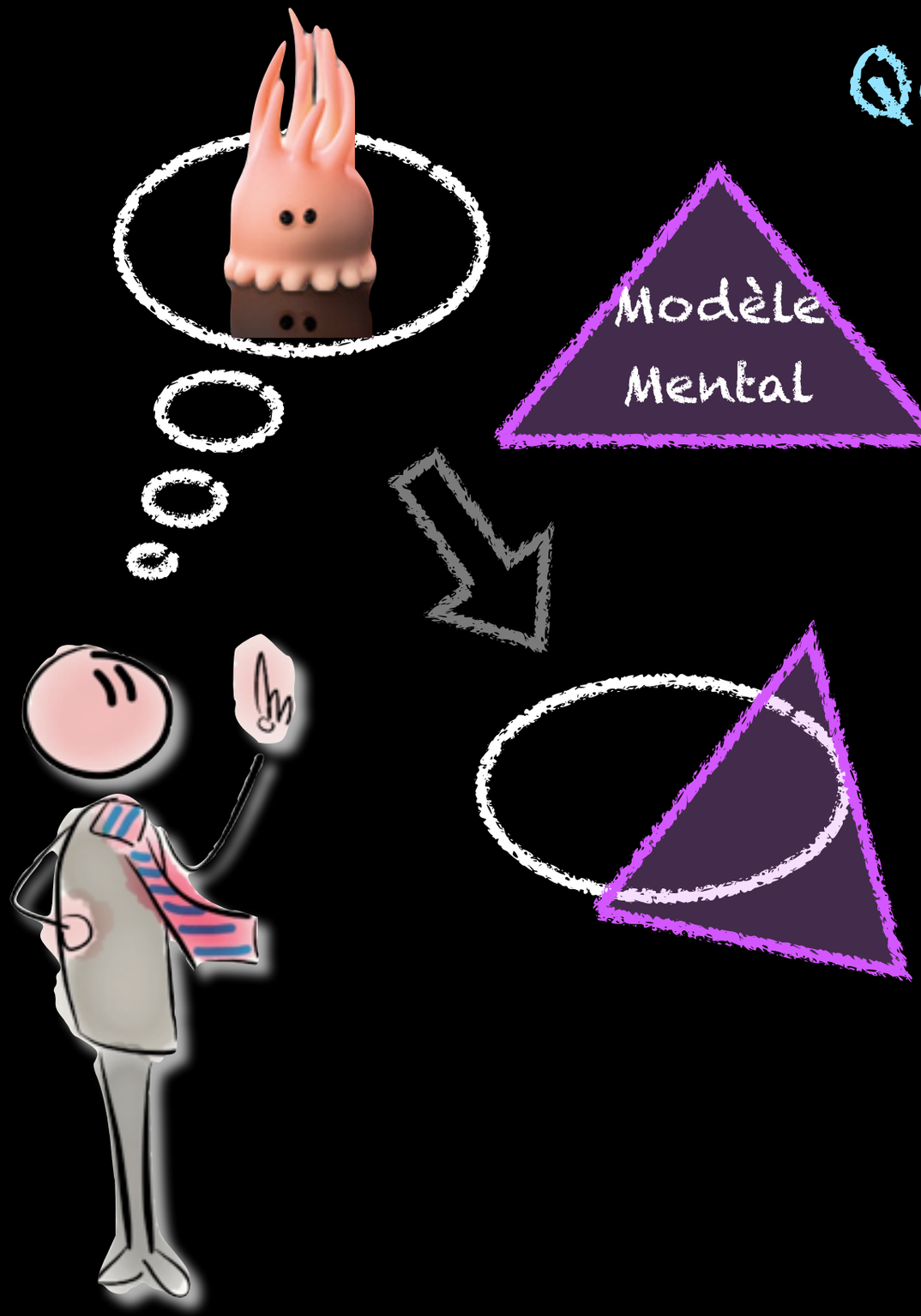
Qu'est ce qui se passe?

Qu'est ce qui se passe?



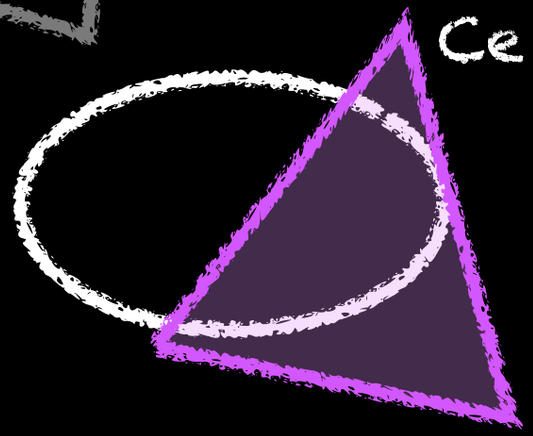
Modèle Mental

Qu'est ce qui se passe?

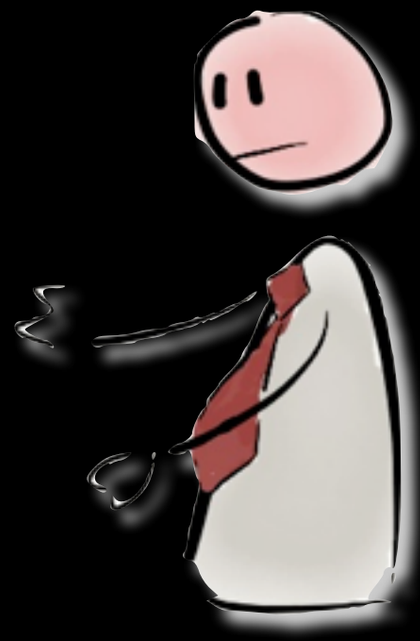


Qu'est ce qui se passe?

Modèle Mental

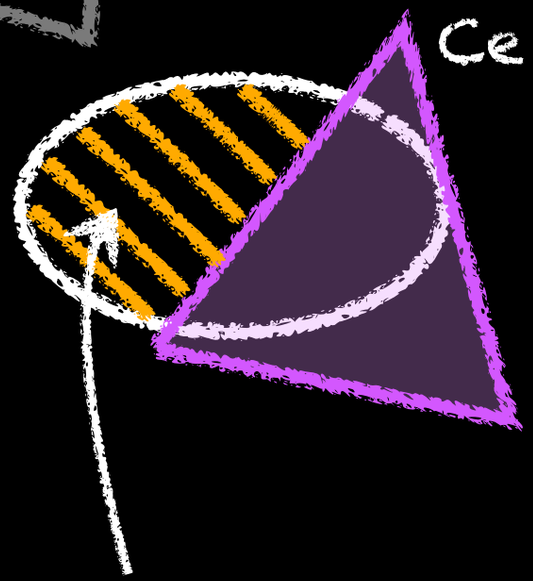


Ce qui est expliqué



Qu'est ce qui se passe?

Modèle Mental



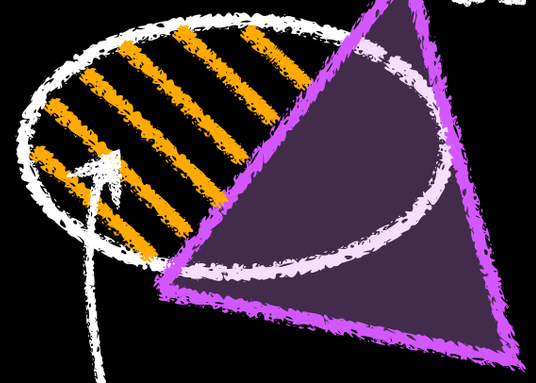
Ce qui est expliqué

Ce qui n'est pas retranscrit

Qu'est ce qui se passe?

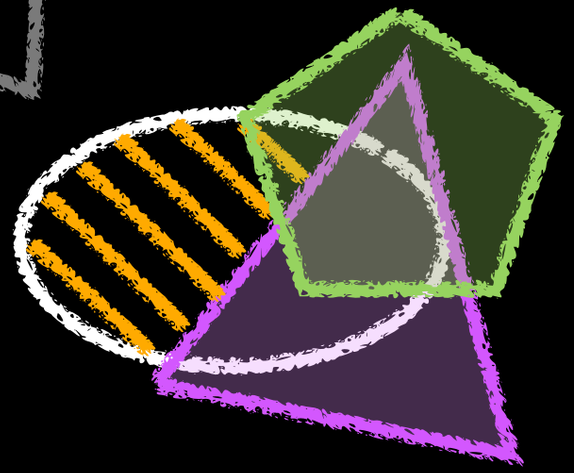
Modèle Mental

Ce qui est expliqué



Ce qui n'est pas retranscrit

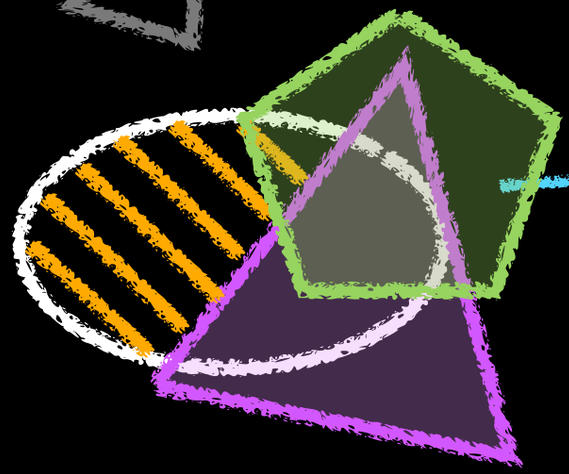
Ce que l'autre comprend

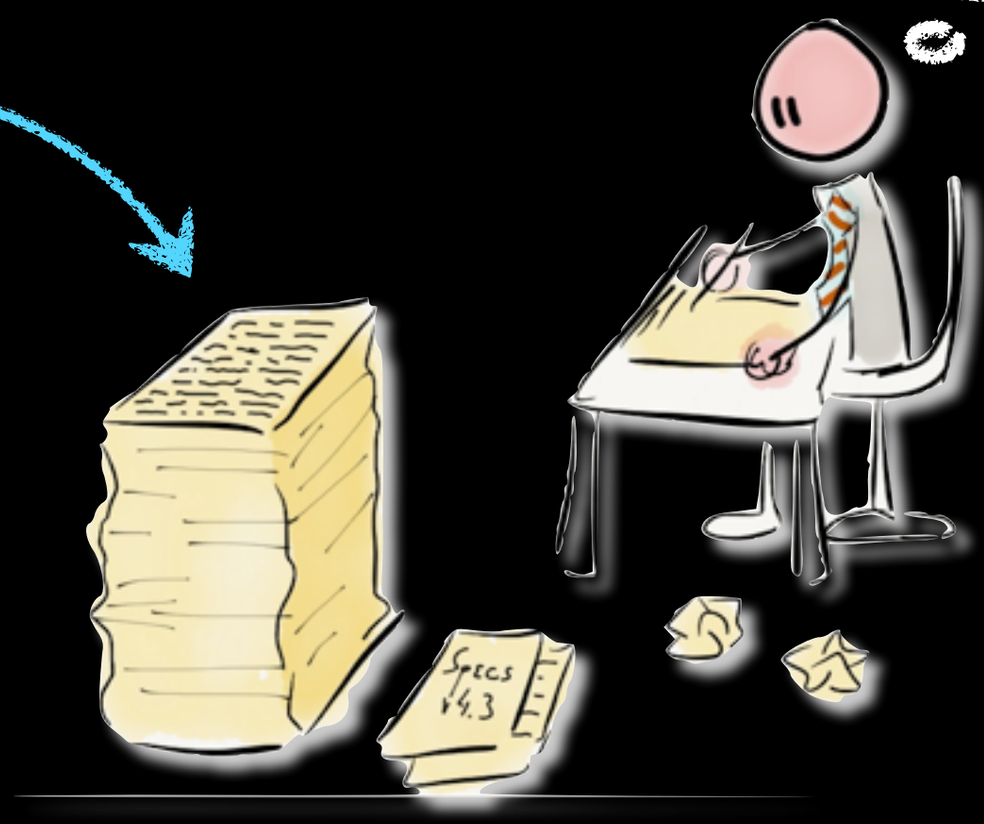


Modèle Mental

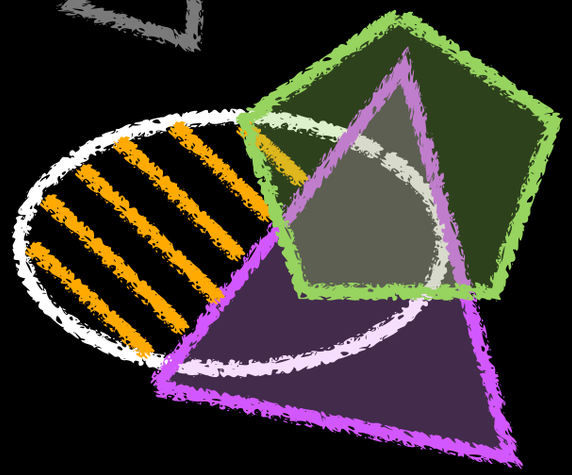
Qu'est ce qui se passe?

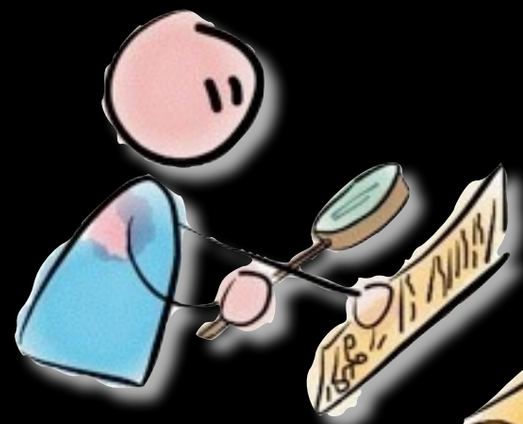
Ce qui est spécifié



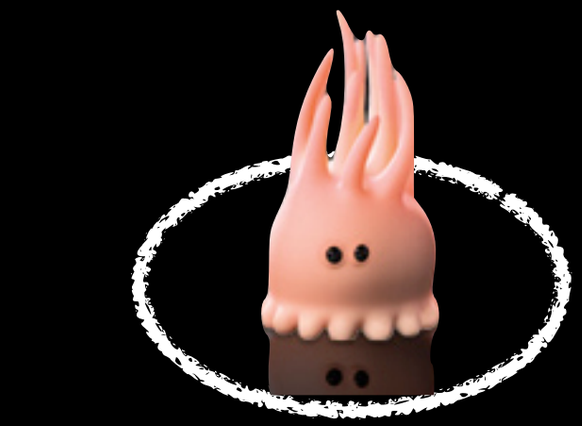


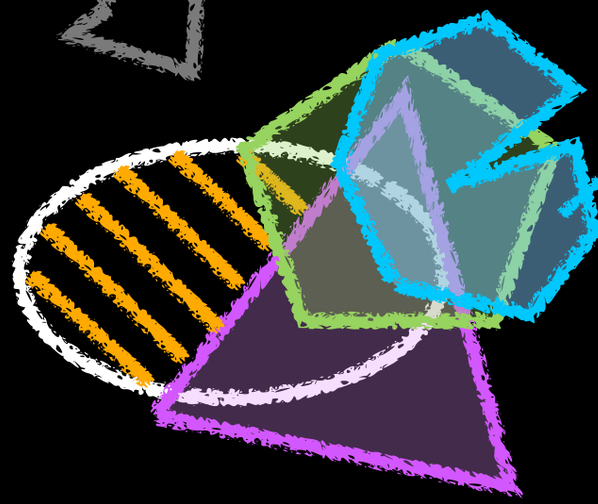
Qu'est ce qui se passe?



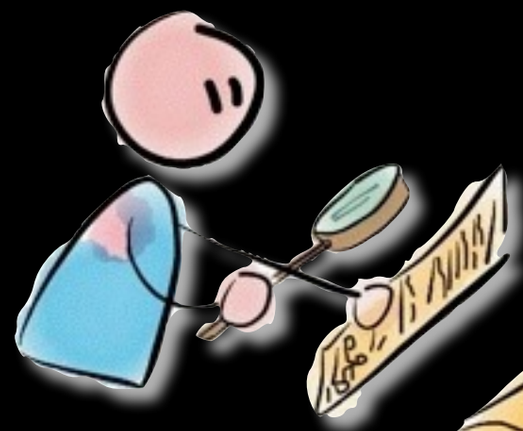


Qu'est ce qui se passe?

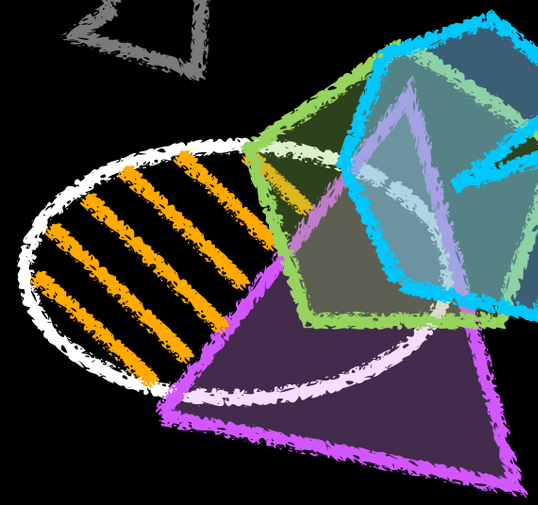




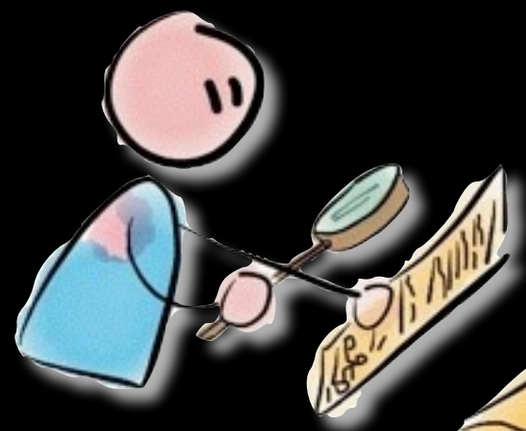
Ce qui va
être testé

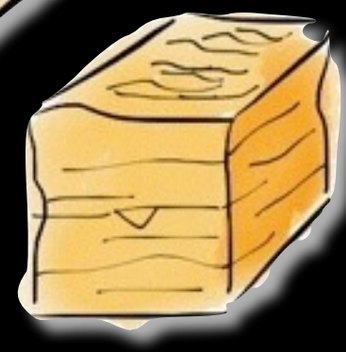


Qu'est ce qui se passe?

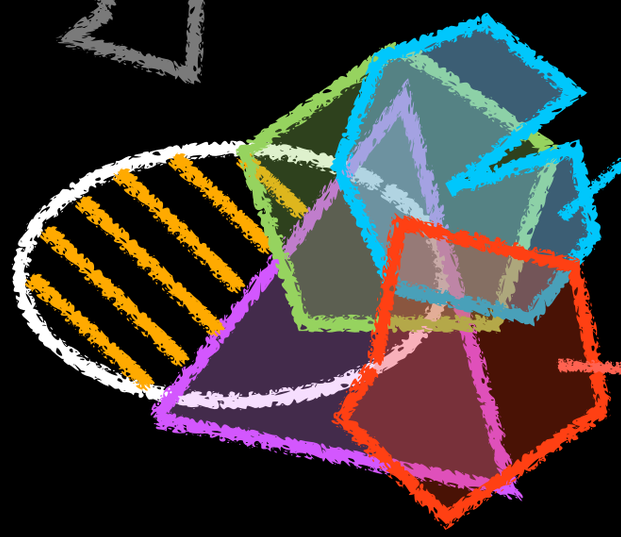


Ce qui va
être testé





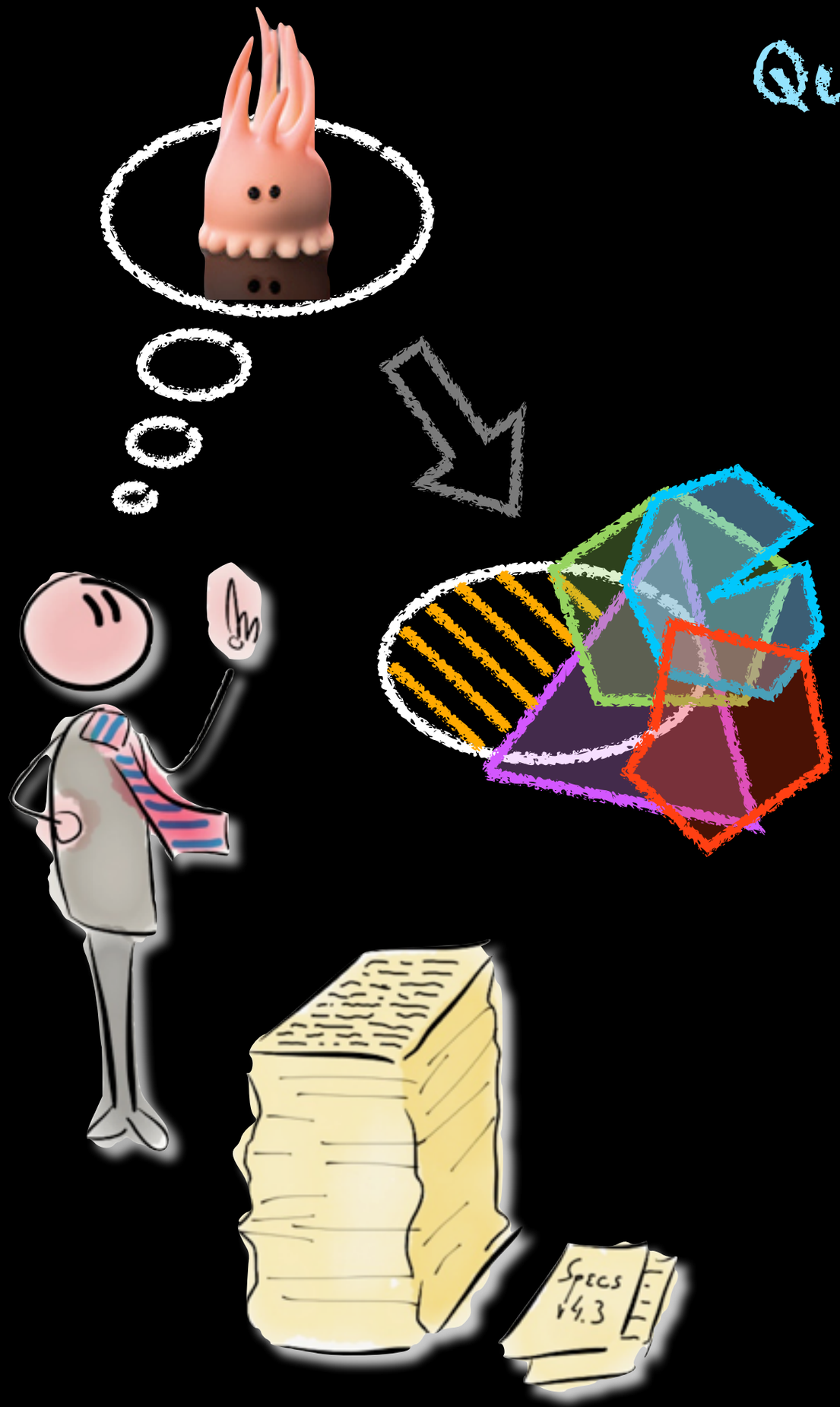
Qu'est ce qui se passe?



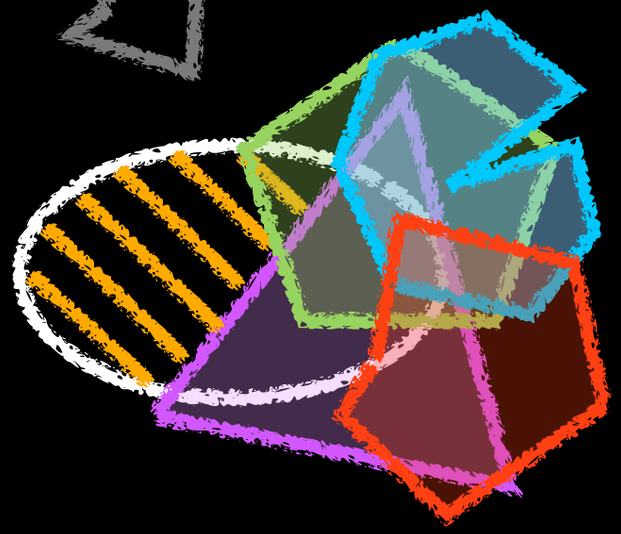
Ce qui va
être testé

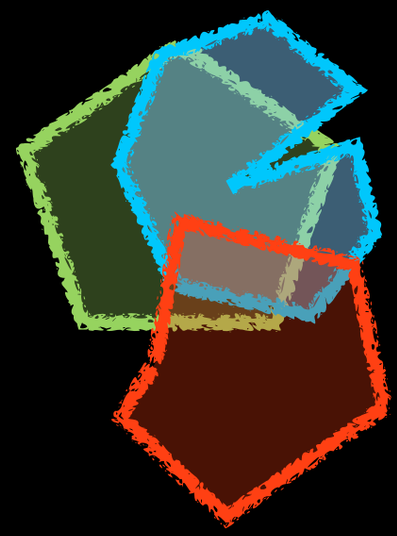
Ce qui est
réalisé

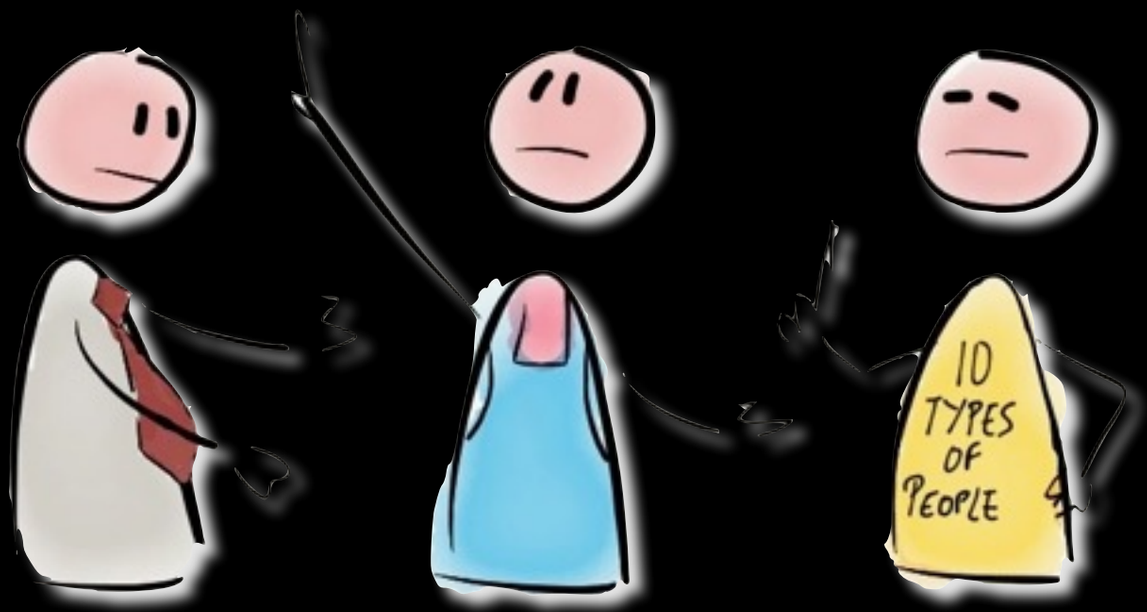
Qu'est ce qui se passe?



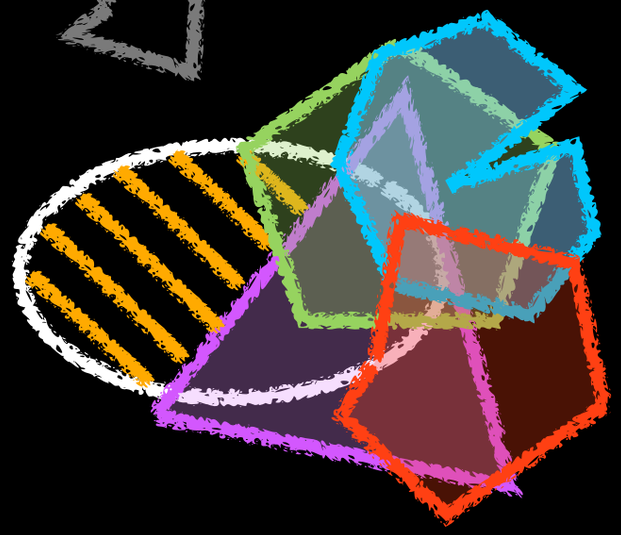
Qu'est ce qui se passe?

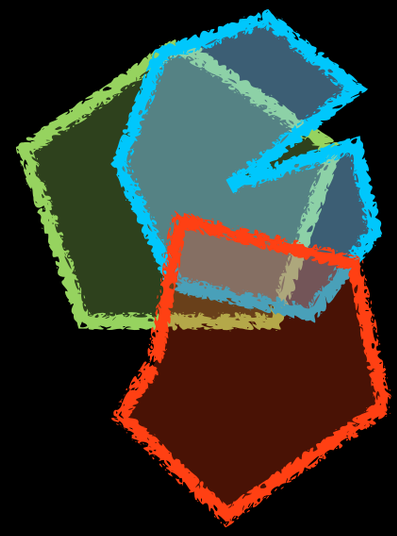


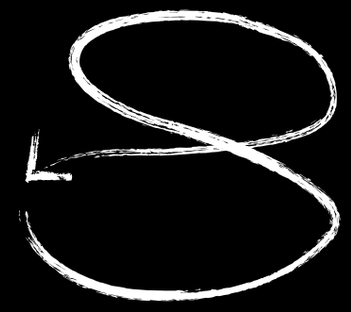


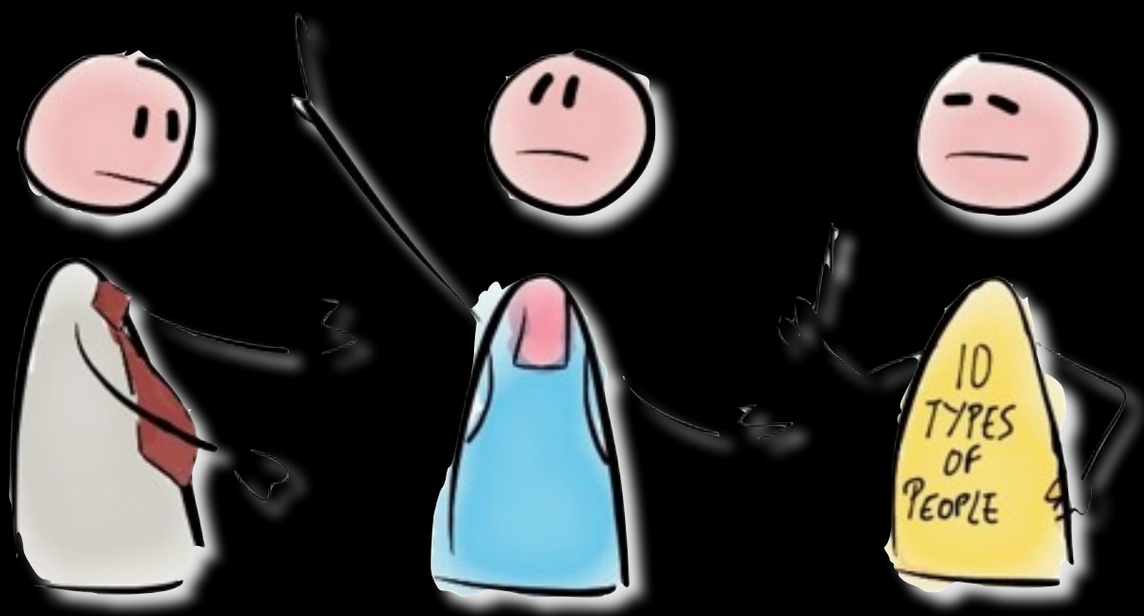


Qu'est ce qui se passe?

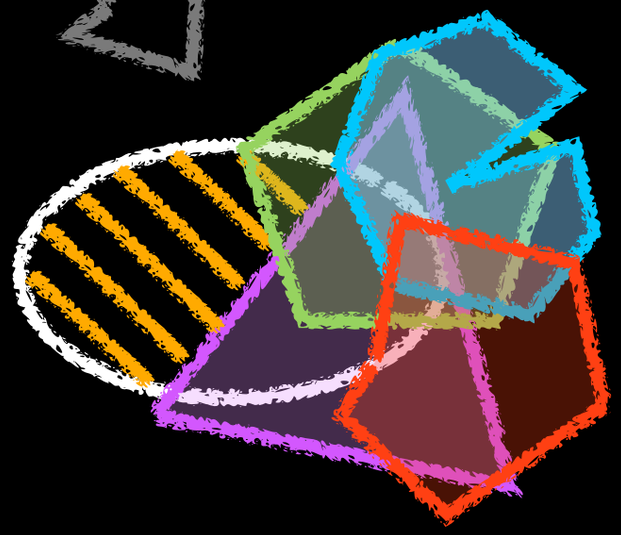


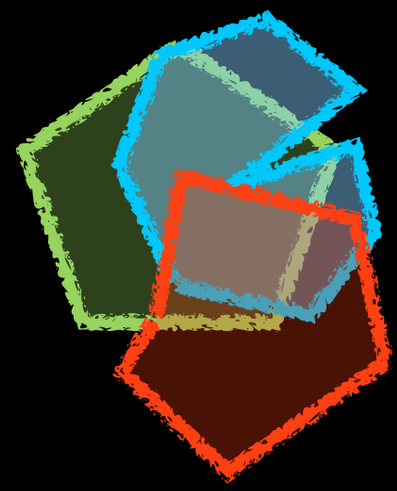


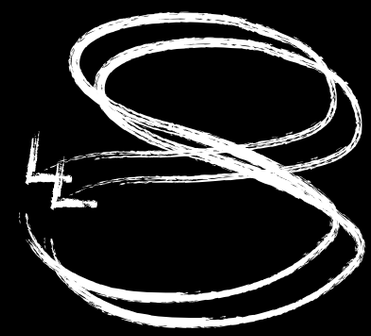


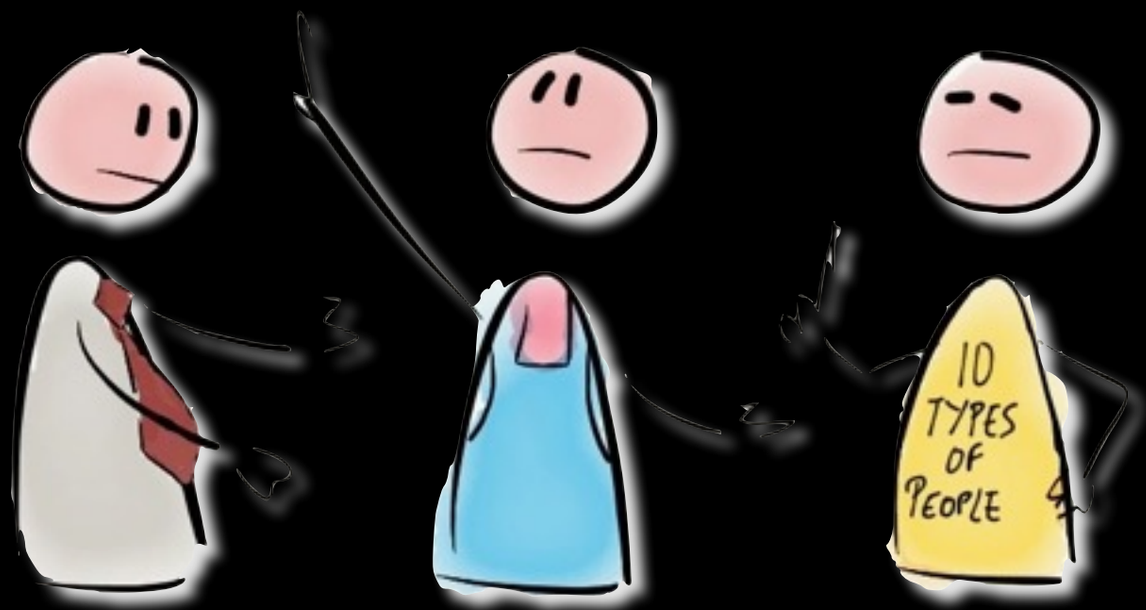


Qu'est ce qui se passe?

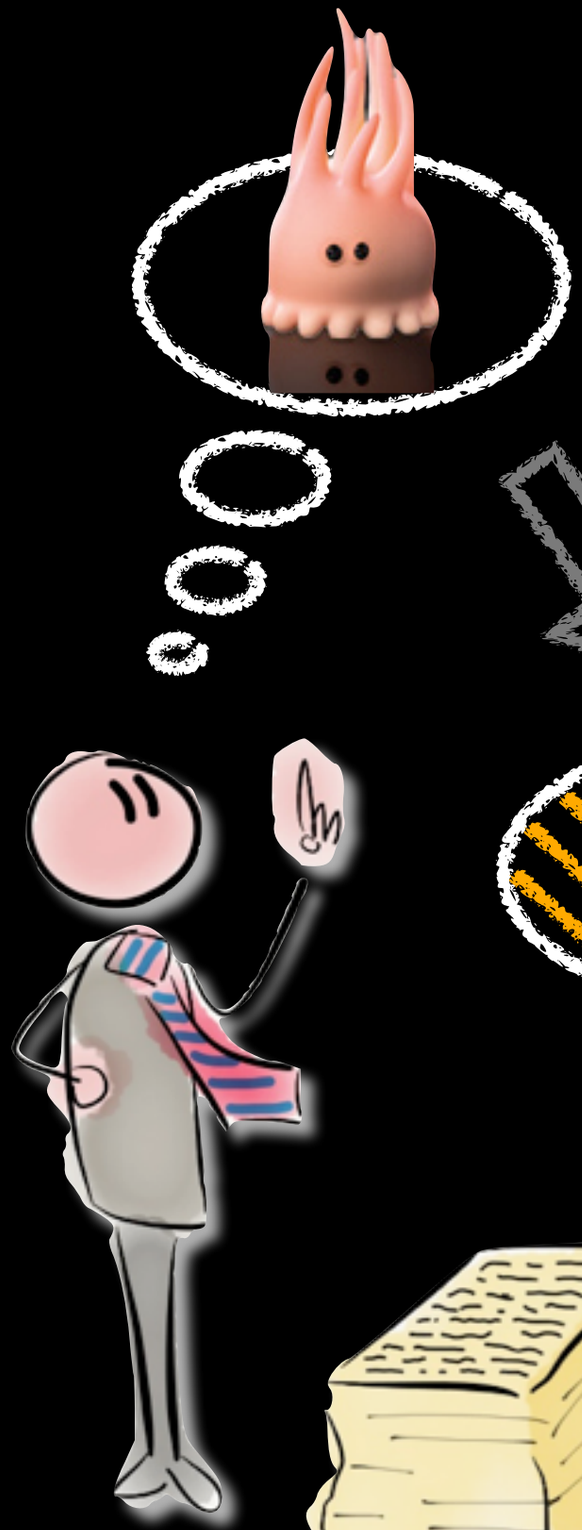


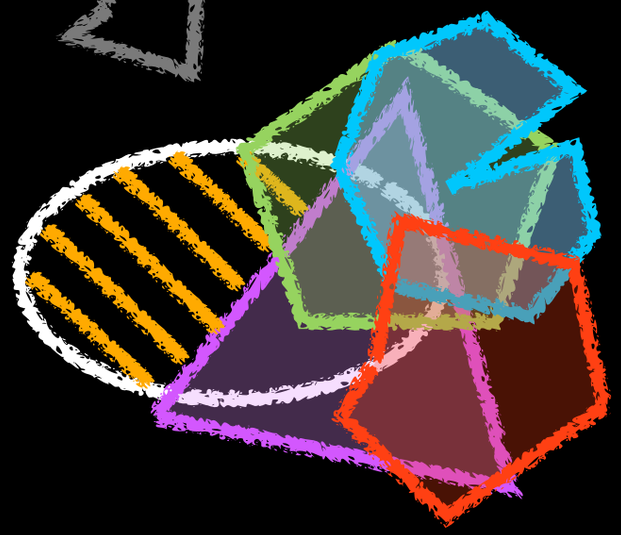


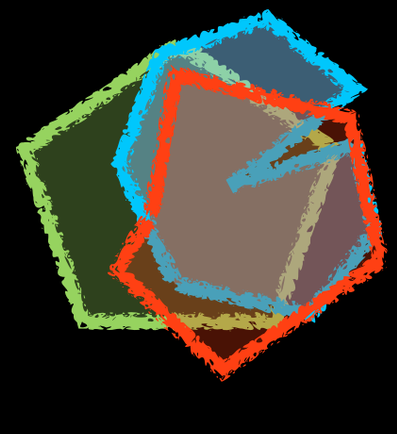


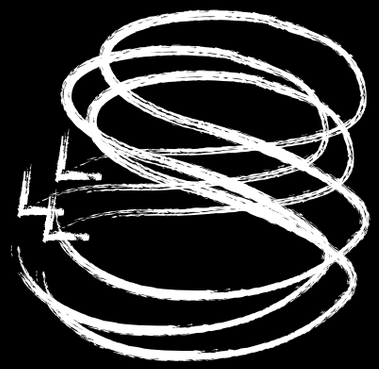


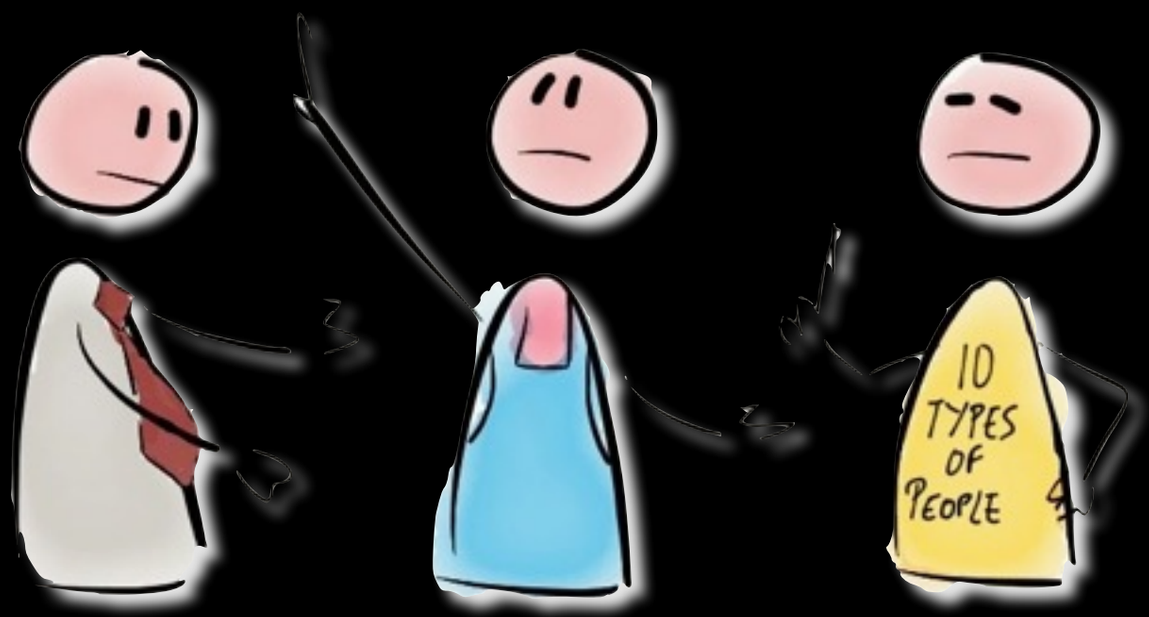
Qu'est ce qui se passe?



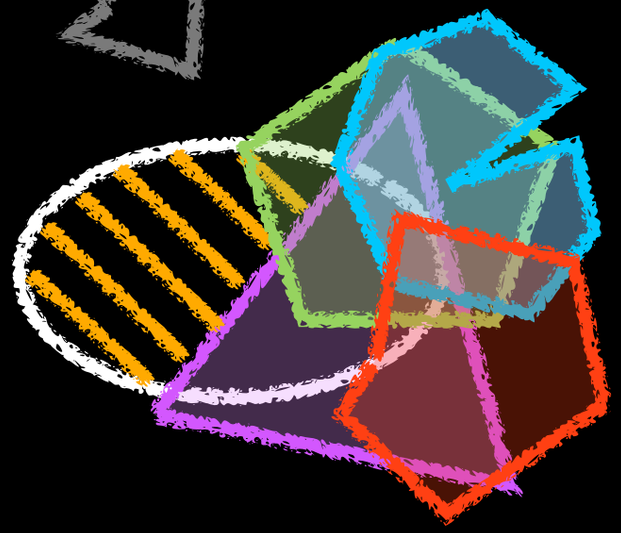


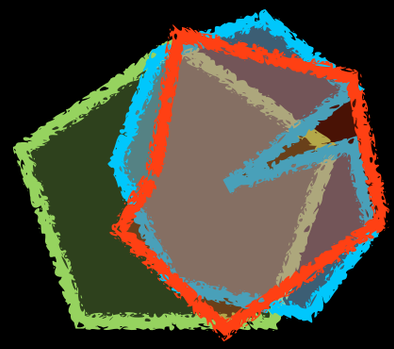


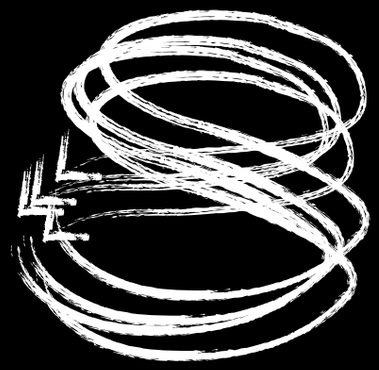


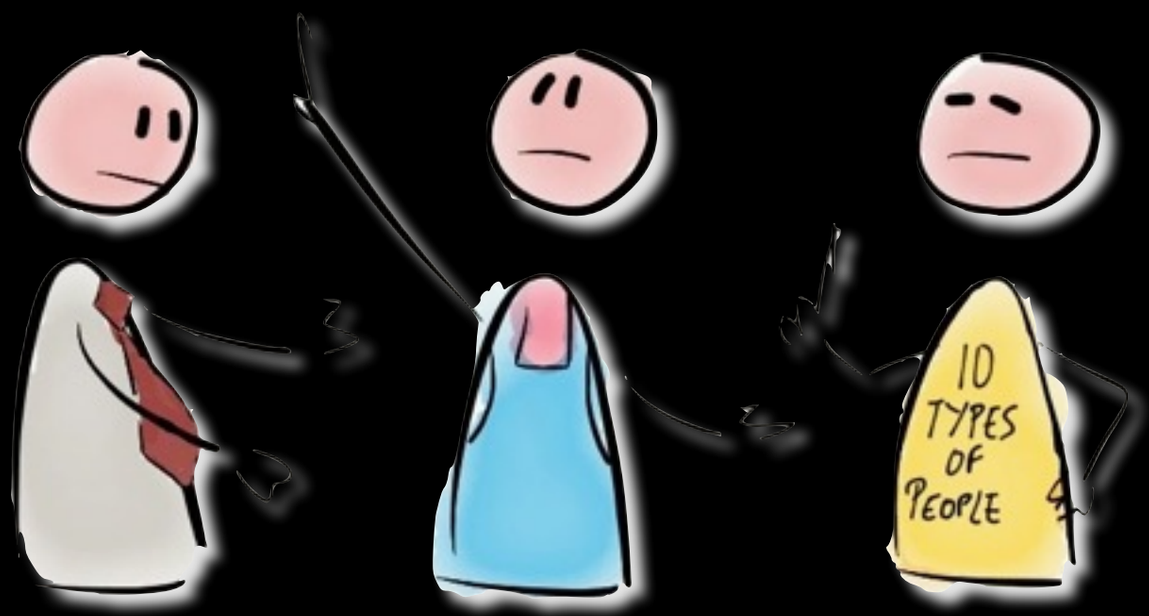


Qu'est ce qui se passe?

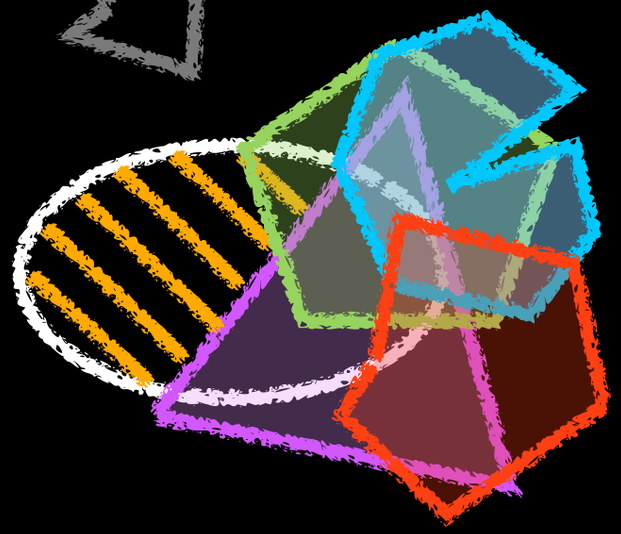


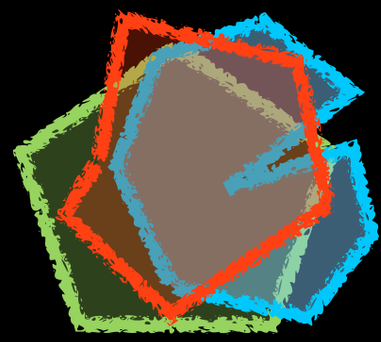


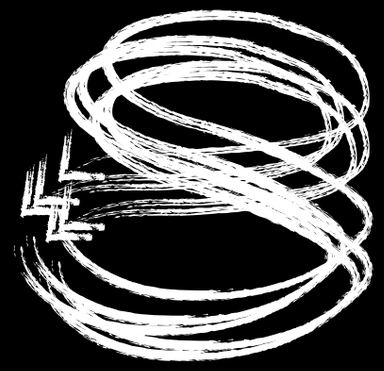


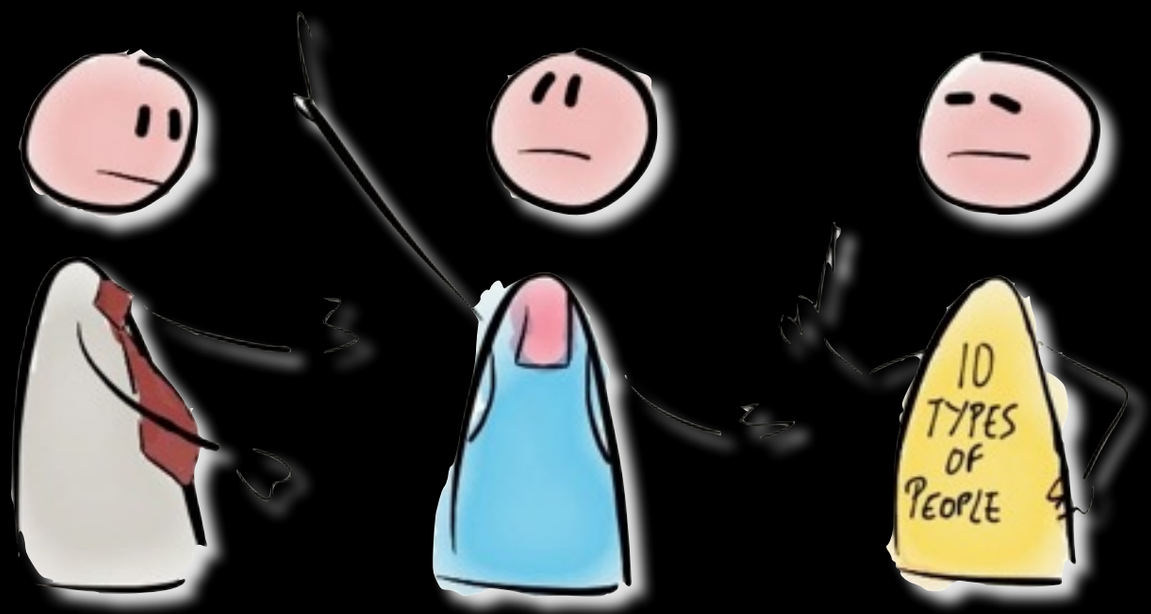


Qu'est ce qui se passe?

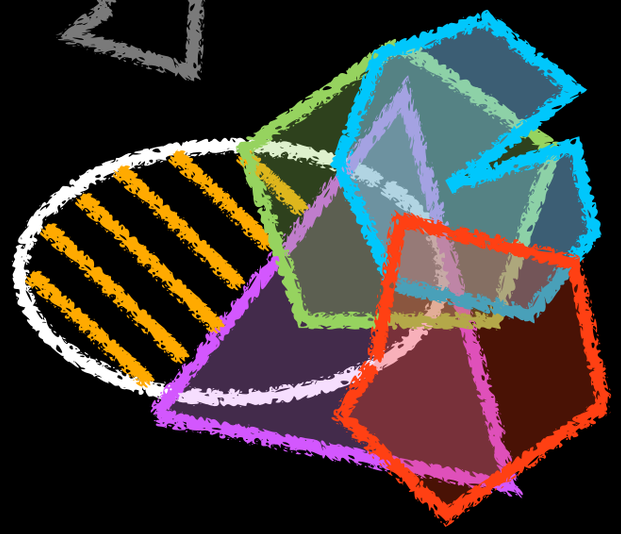


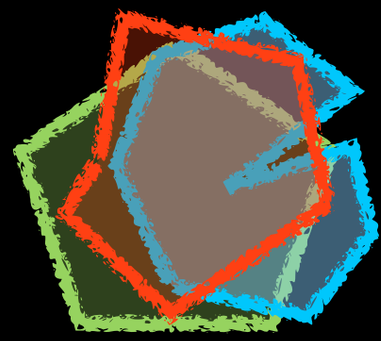


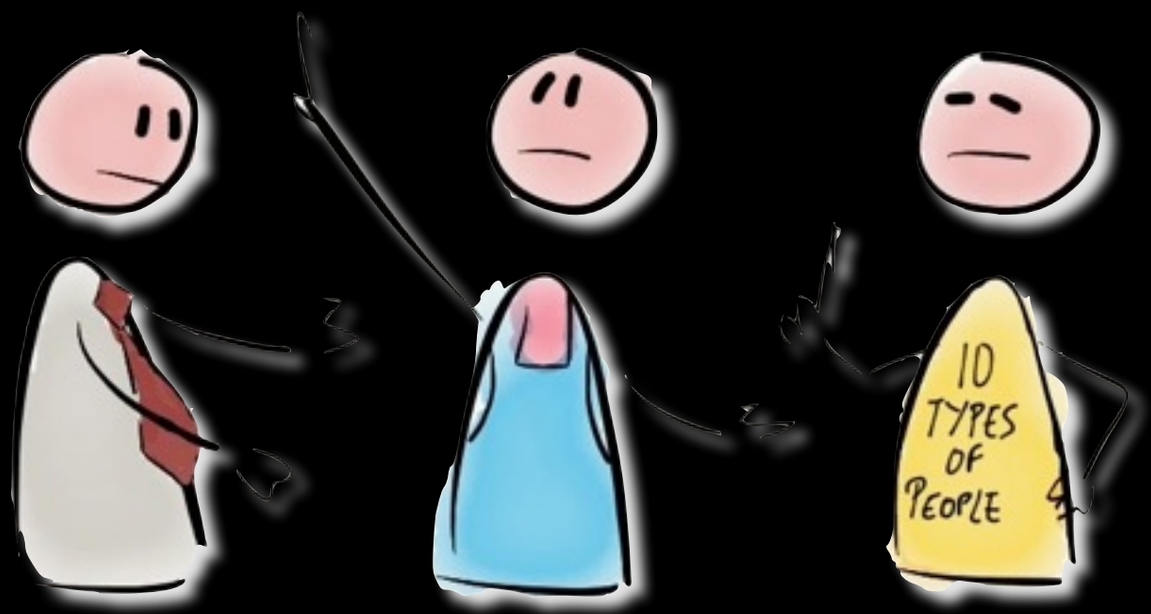




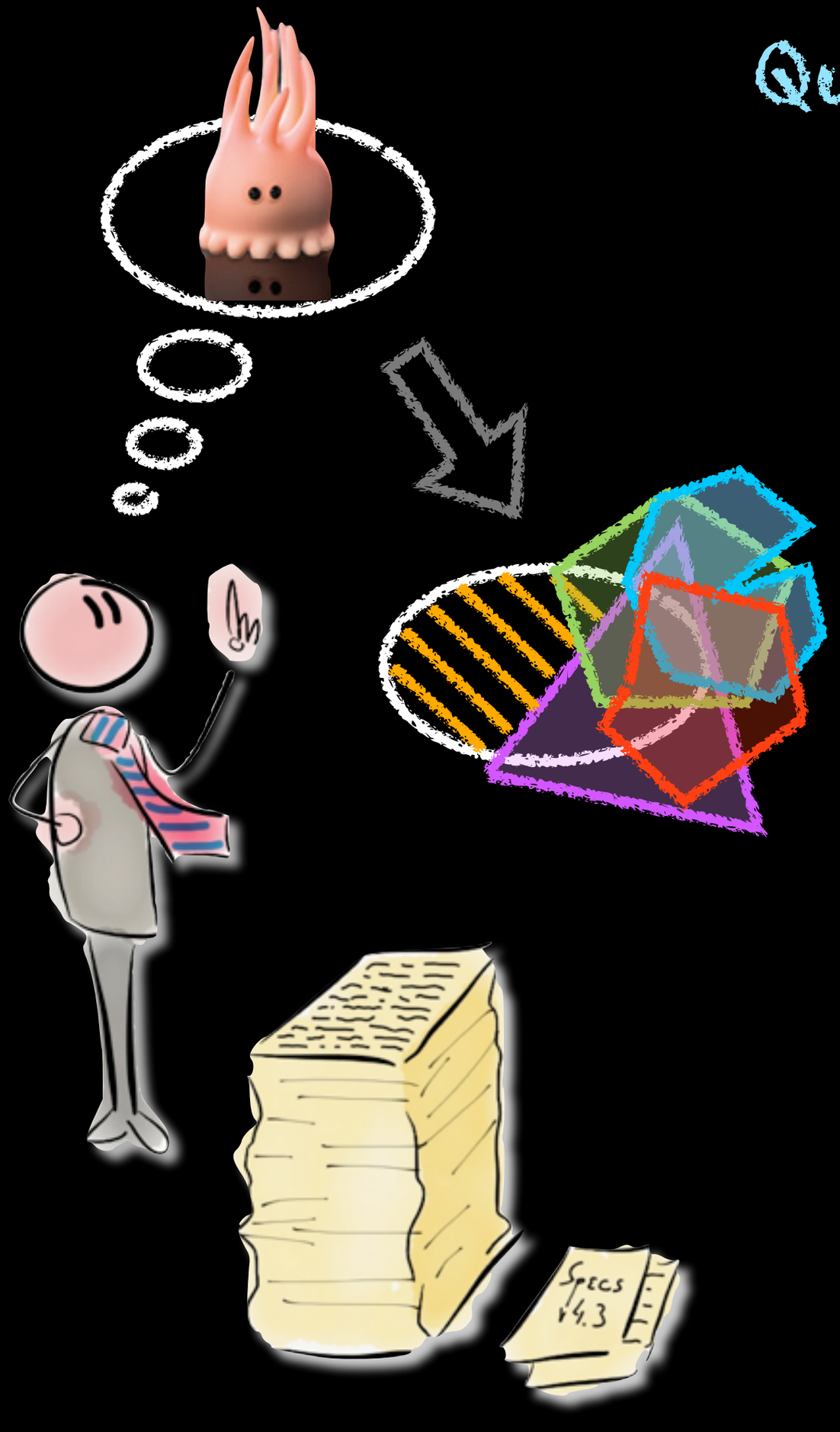
Qu'est ce qui se passe?



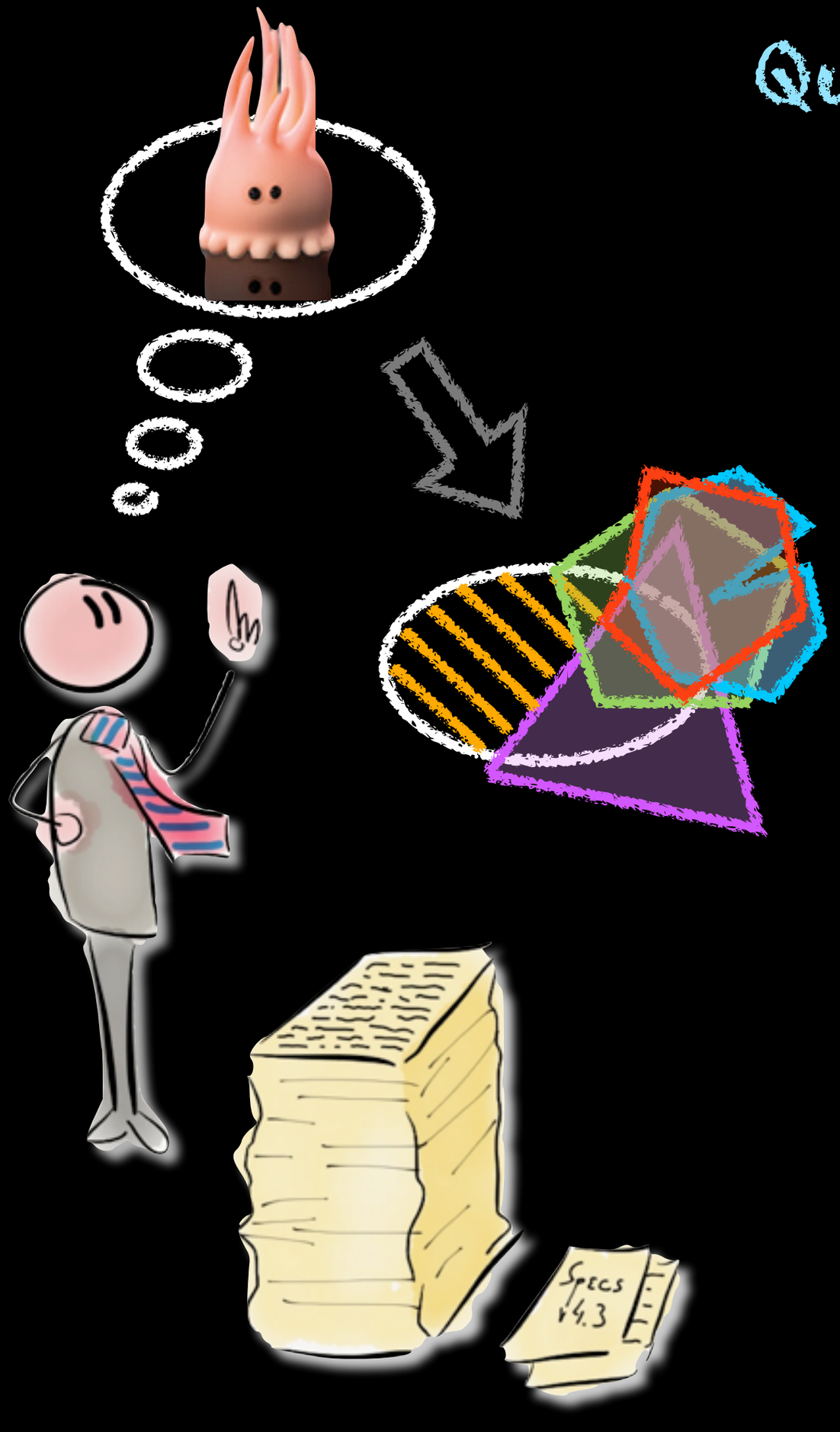




Qu'est ce qui se passe?

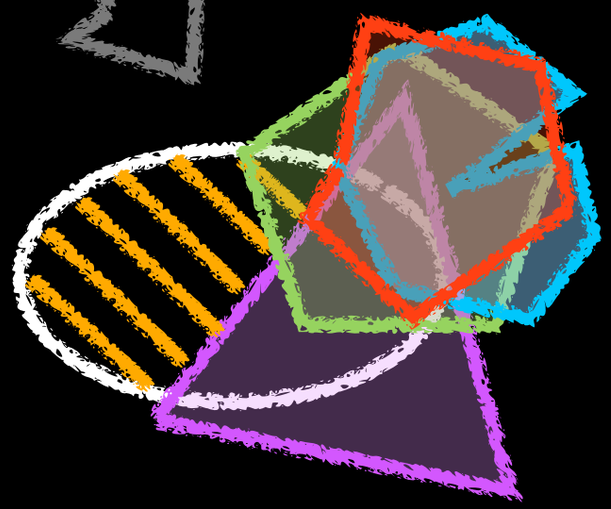


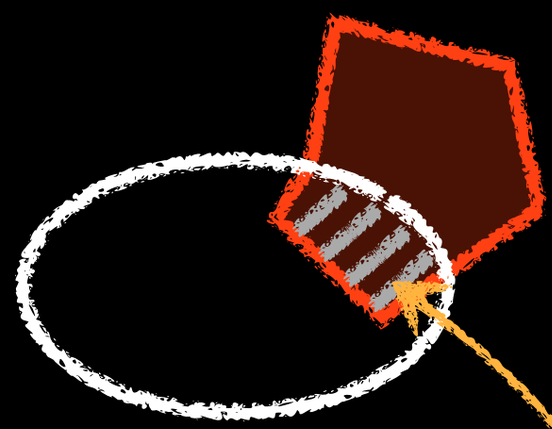
Qu'est ce qui se passe?



Qu'est ce qui se passe?

Ce qui a été réalisé

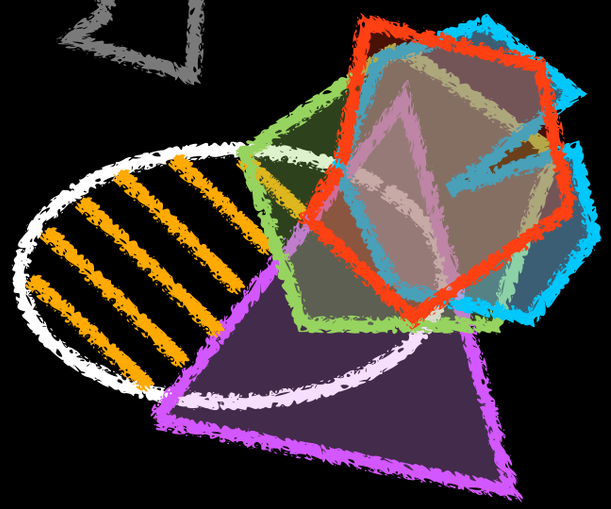


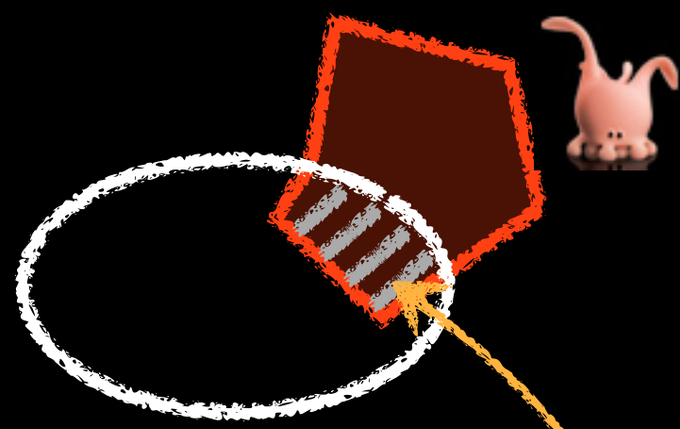


Correspond à l'idée initiale

Qu'est ce qui se passe?

Ce qui a été réalisé



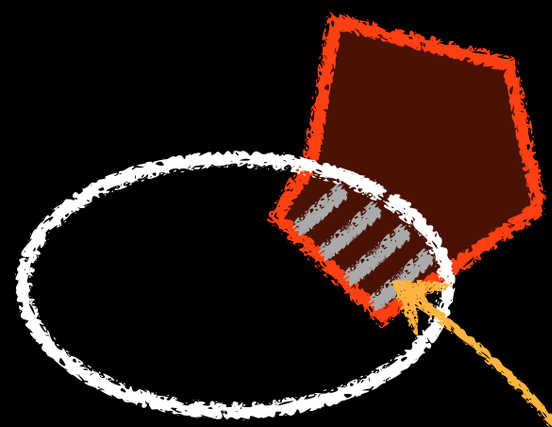


Correspond à l'idée initiale

Qu'est ce qui se passe?

Ce qui a été réalisé

VS



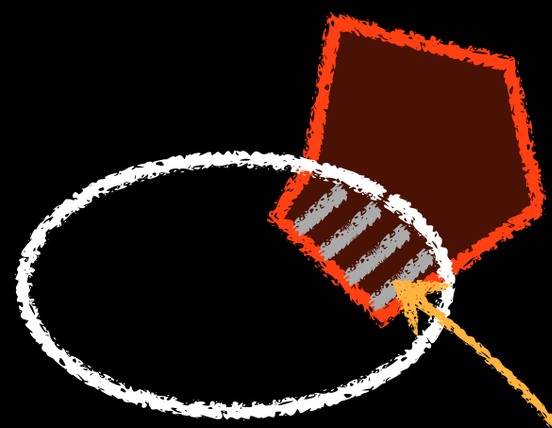
Correspond à l'idée initiale

Qu'est ce qui se passe?

NNNAaannnnnn

VS

Ce qui a été réalisé



Correspond à l'idée initiale

"is this a bug or is this not a bug

"is this a bug or is this not a bug a bug?

"is this a bug or is this not a bug

a bug?

a feature?

"is this a bug or is this not a bug

a bug?

a feature?

a beature?

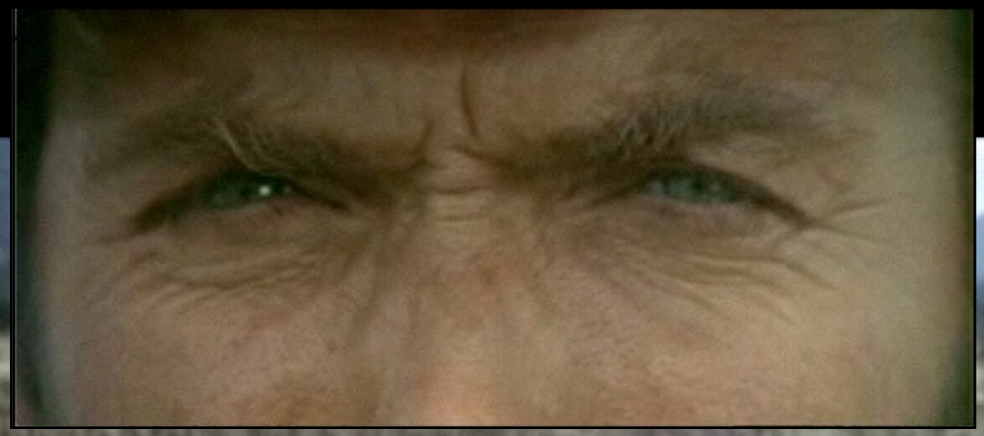
"is this a bug or is this not a bug

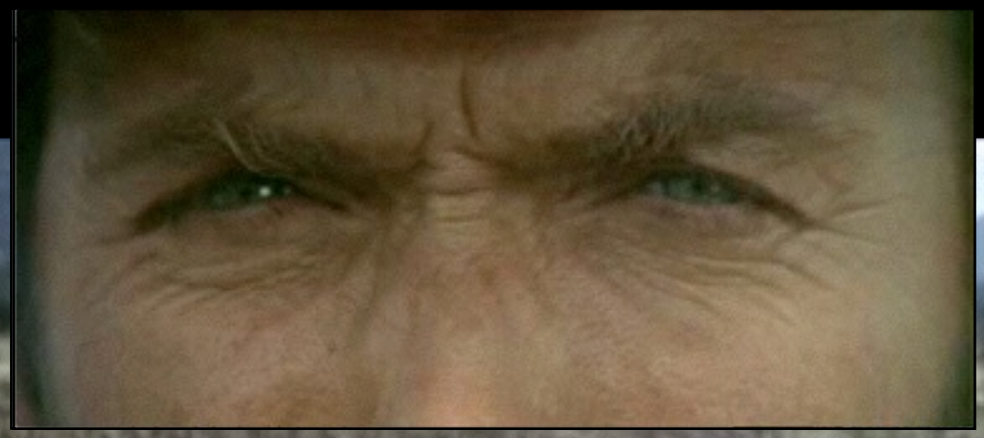
a bug?

a feature?

a beature?

an enhancement?





3 steps

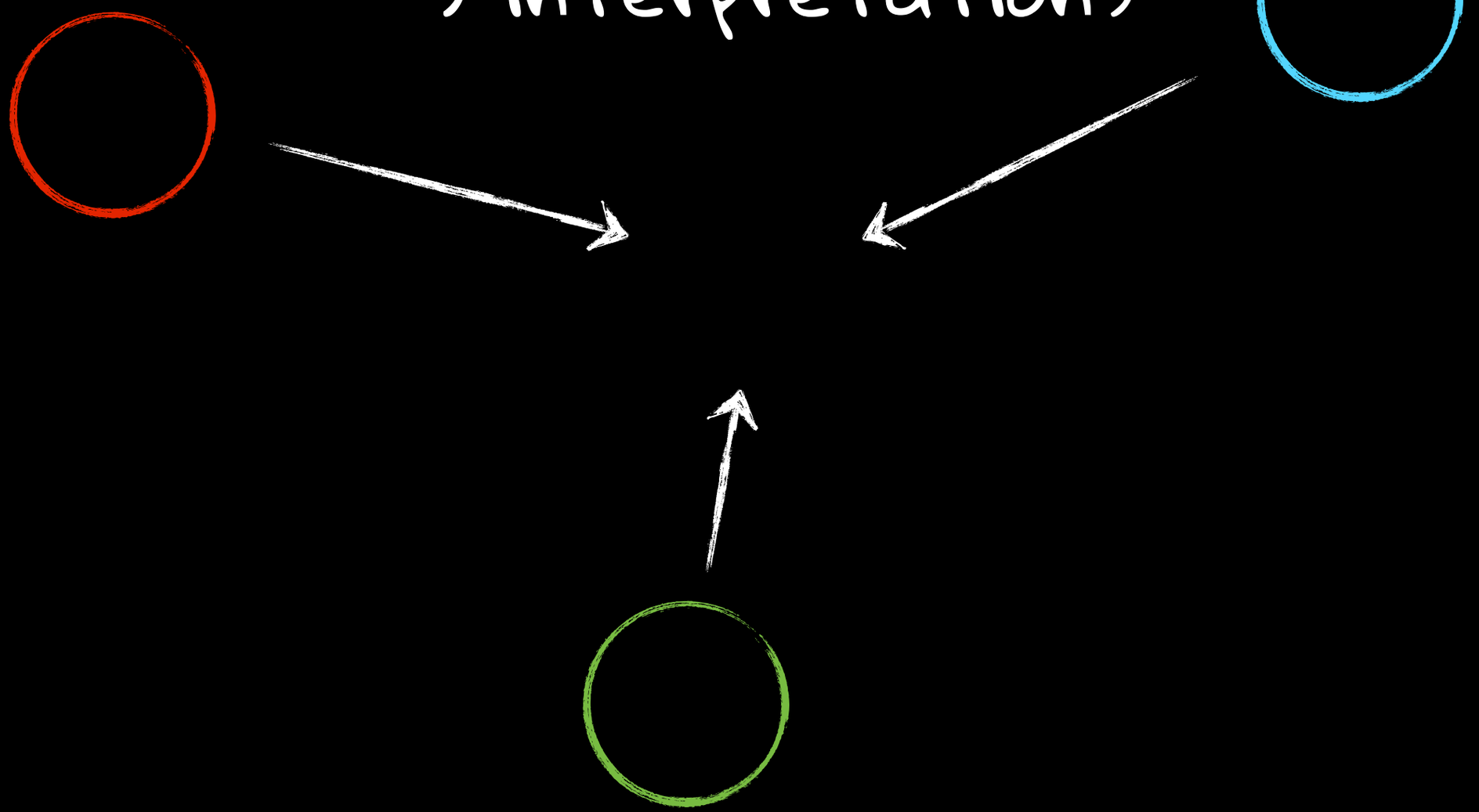
3 model

3 interpretations

3 steps

3 model

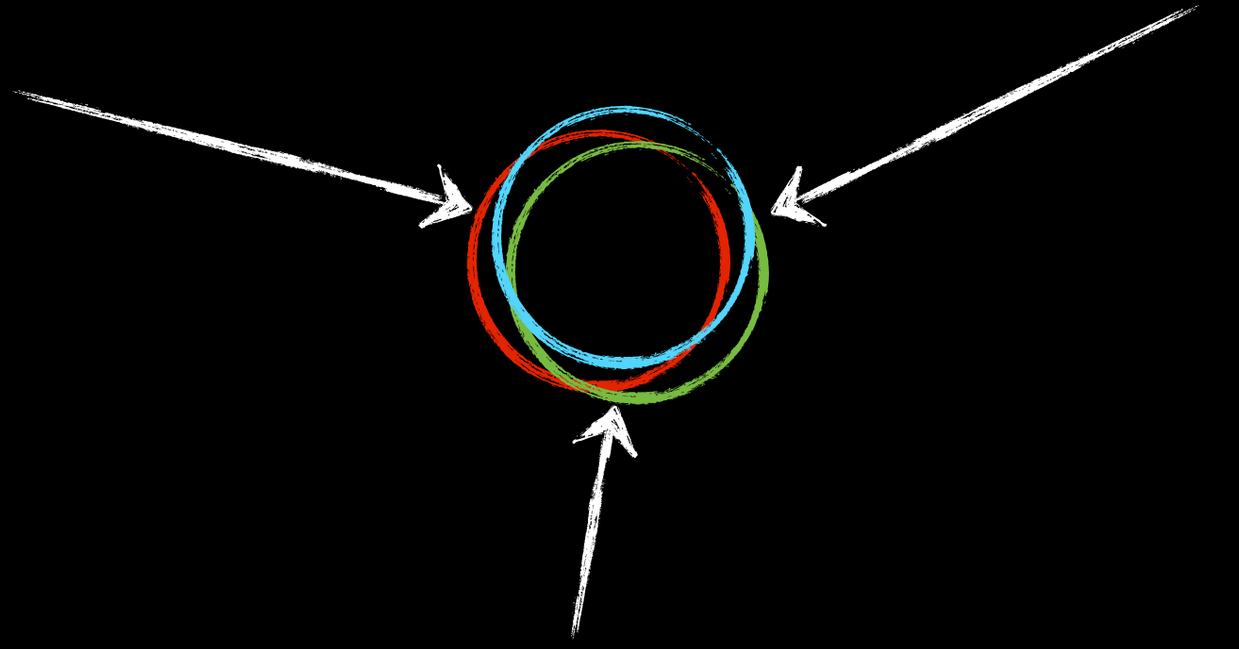
3 interpretations



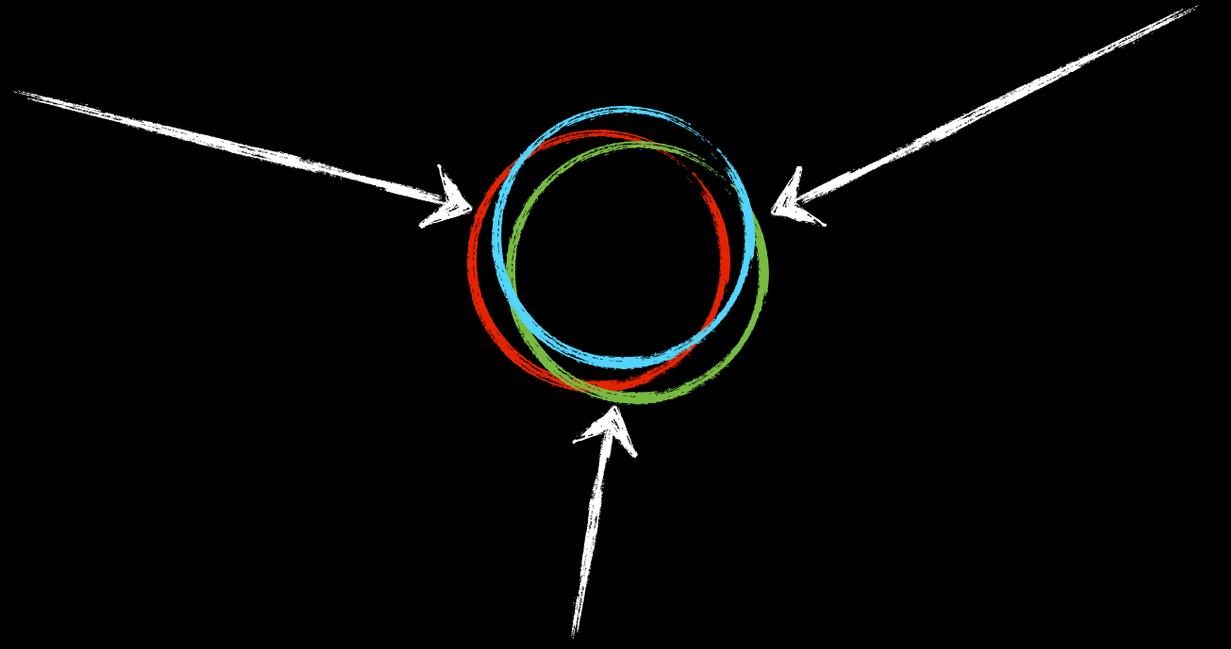
3 steps

3 model

3 interpretations



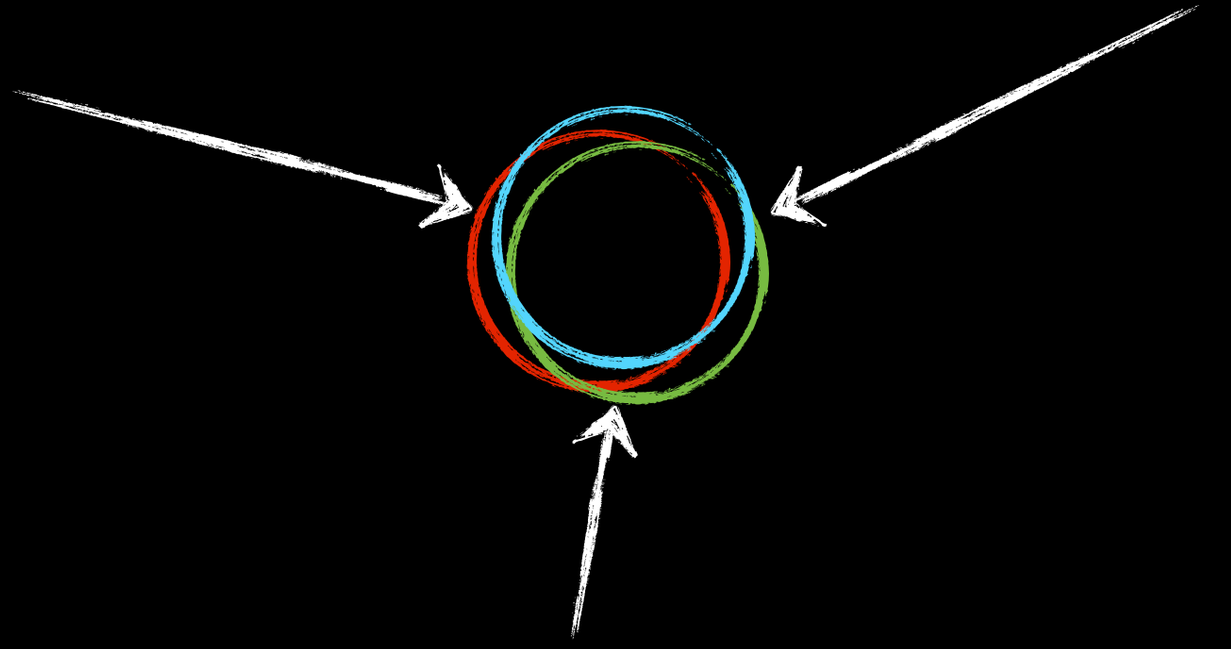

LEGRAPH
MUNTZ ASCENDS
ZIGGURAT
NEWS



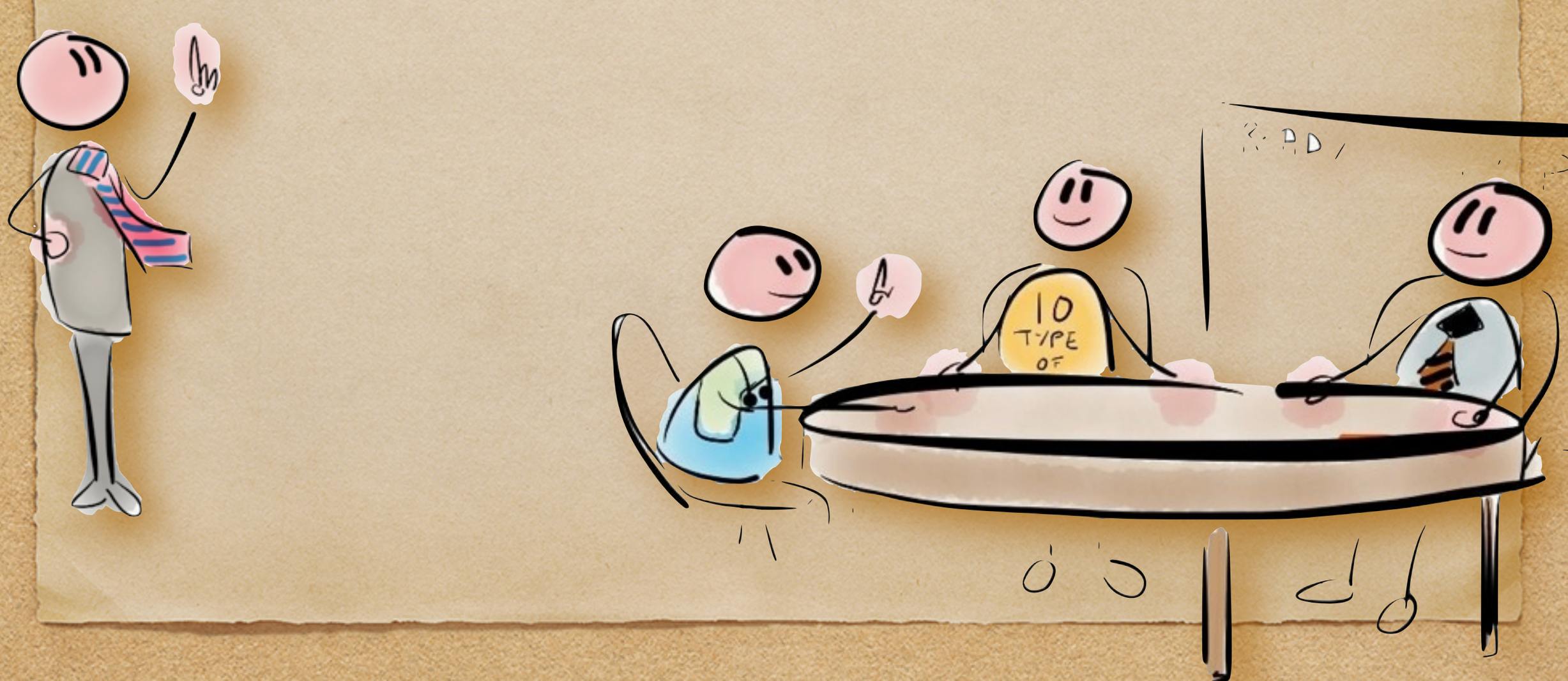
3 visions

1 model

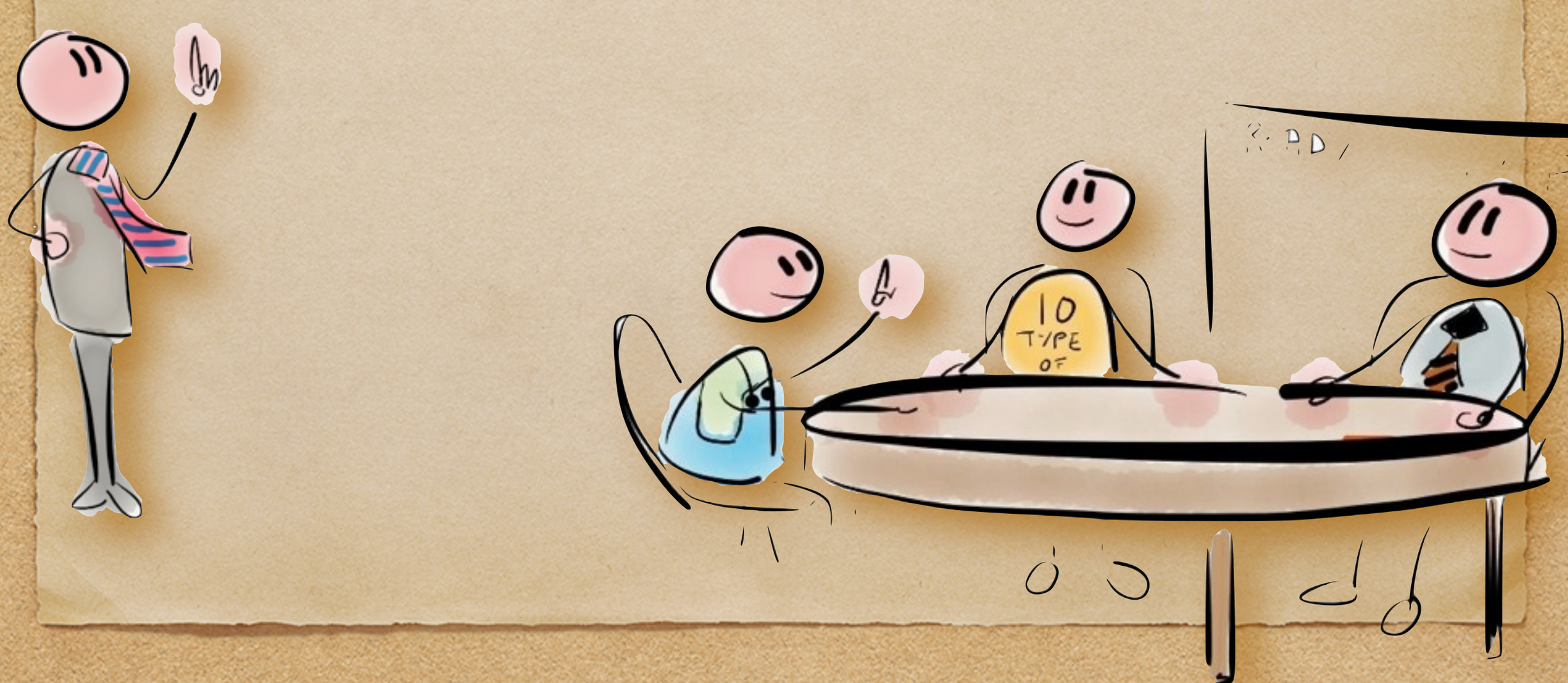
1 goal



Three Amigos


10
TYPE
OF

BDD is about conversation



Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools

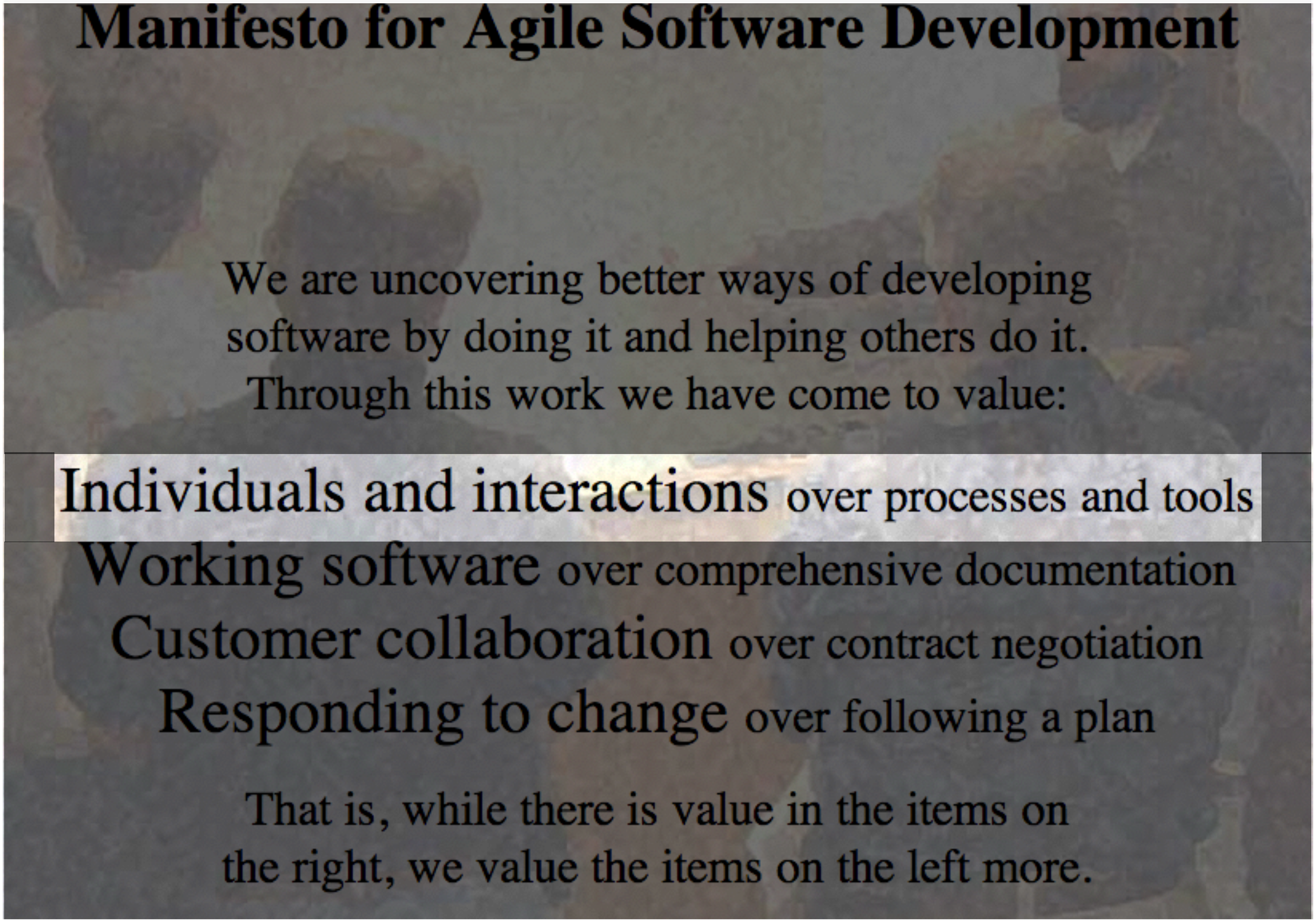
Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

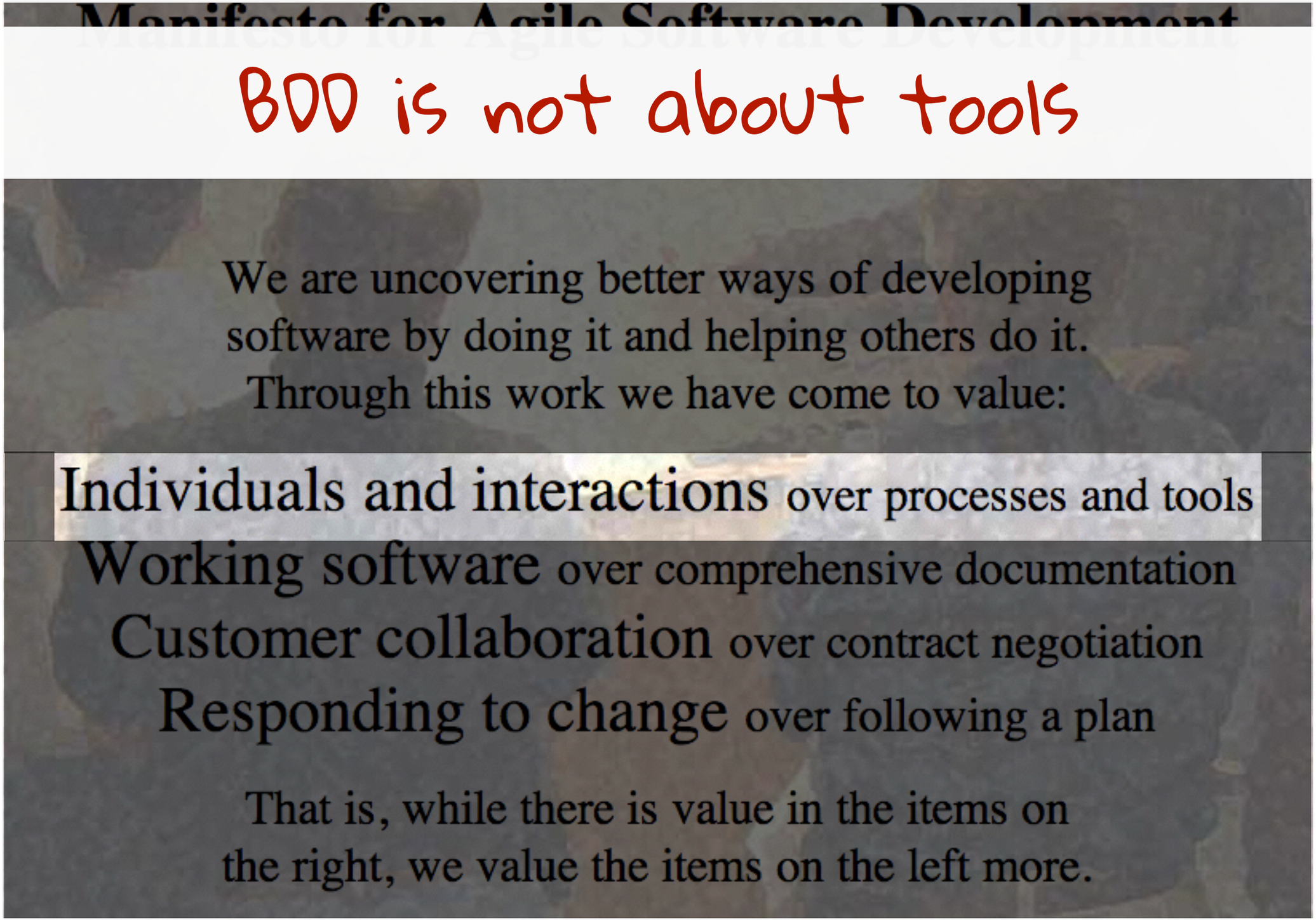
That is, while there is value in the items on
the right, we value the items on the left more.

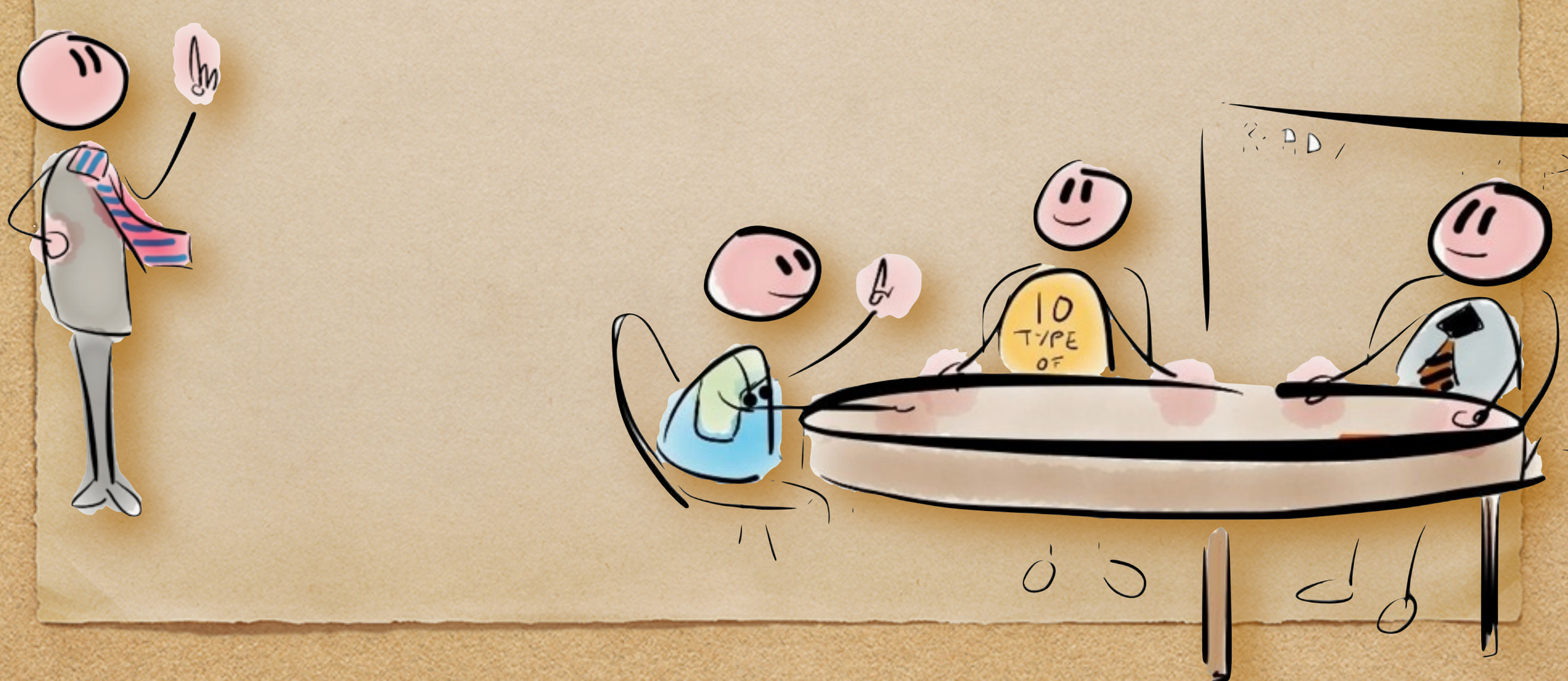
BDD is about conversation



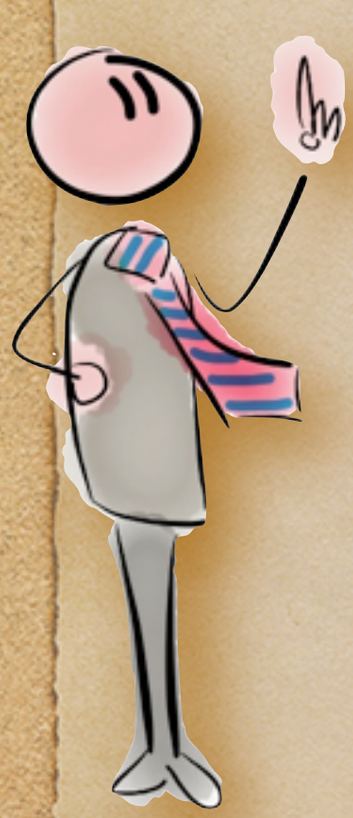
BDD is about conversation

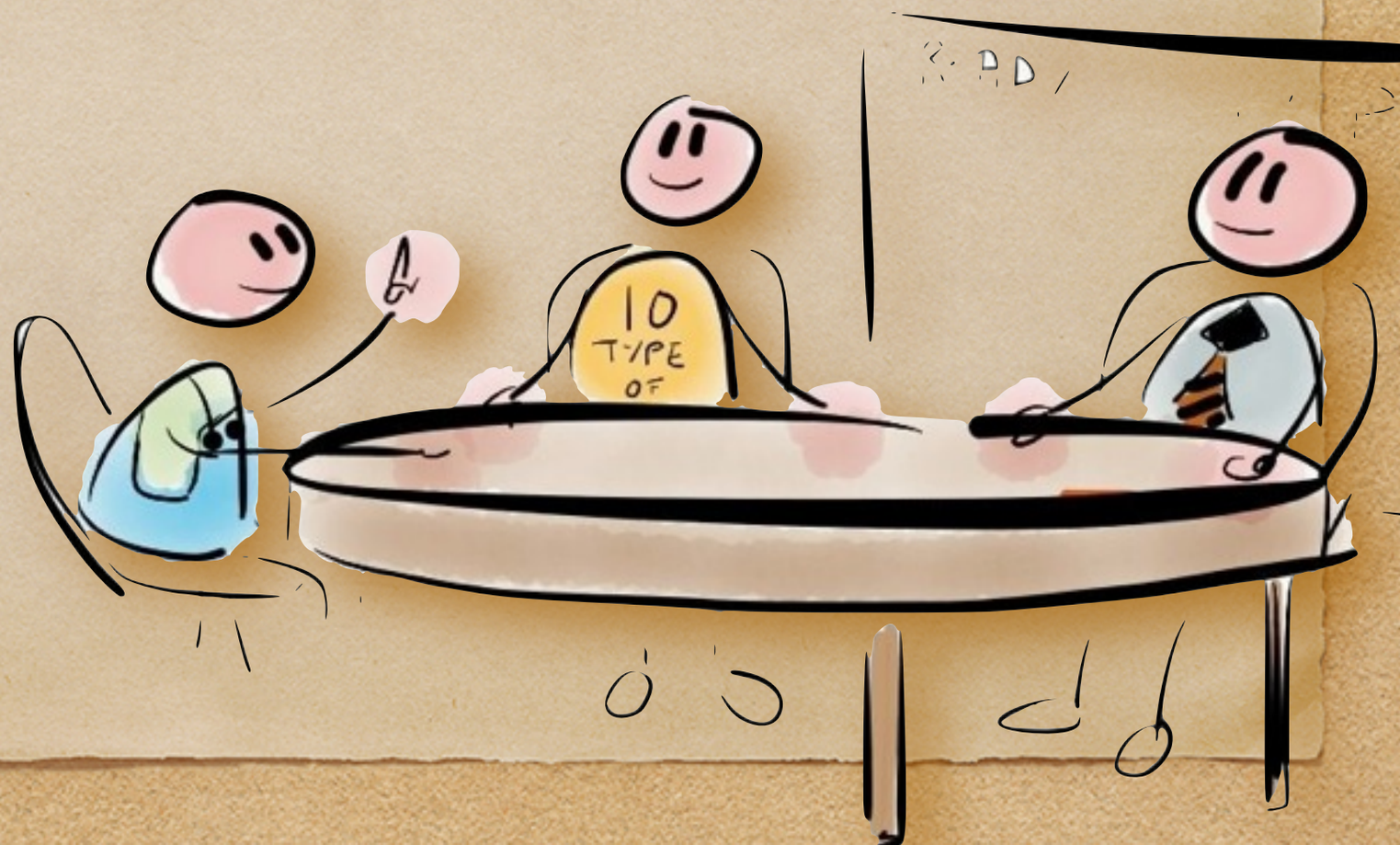
BDD is not about tools




10
TYPE
OF

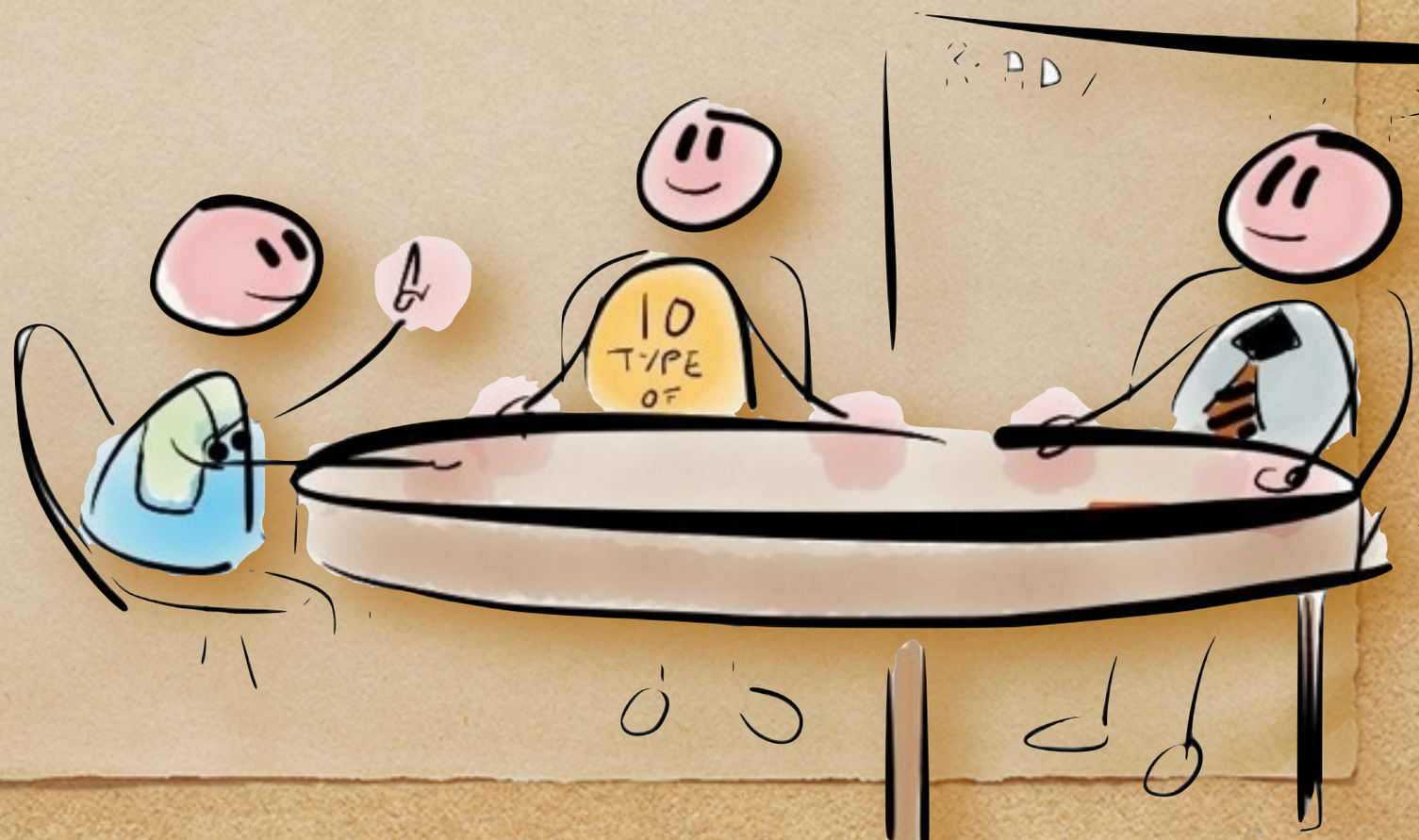
What?





What?

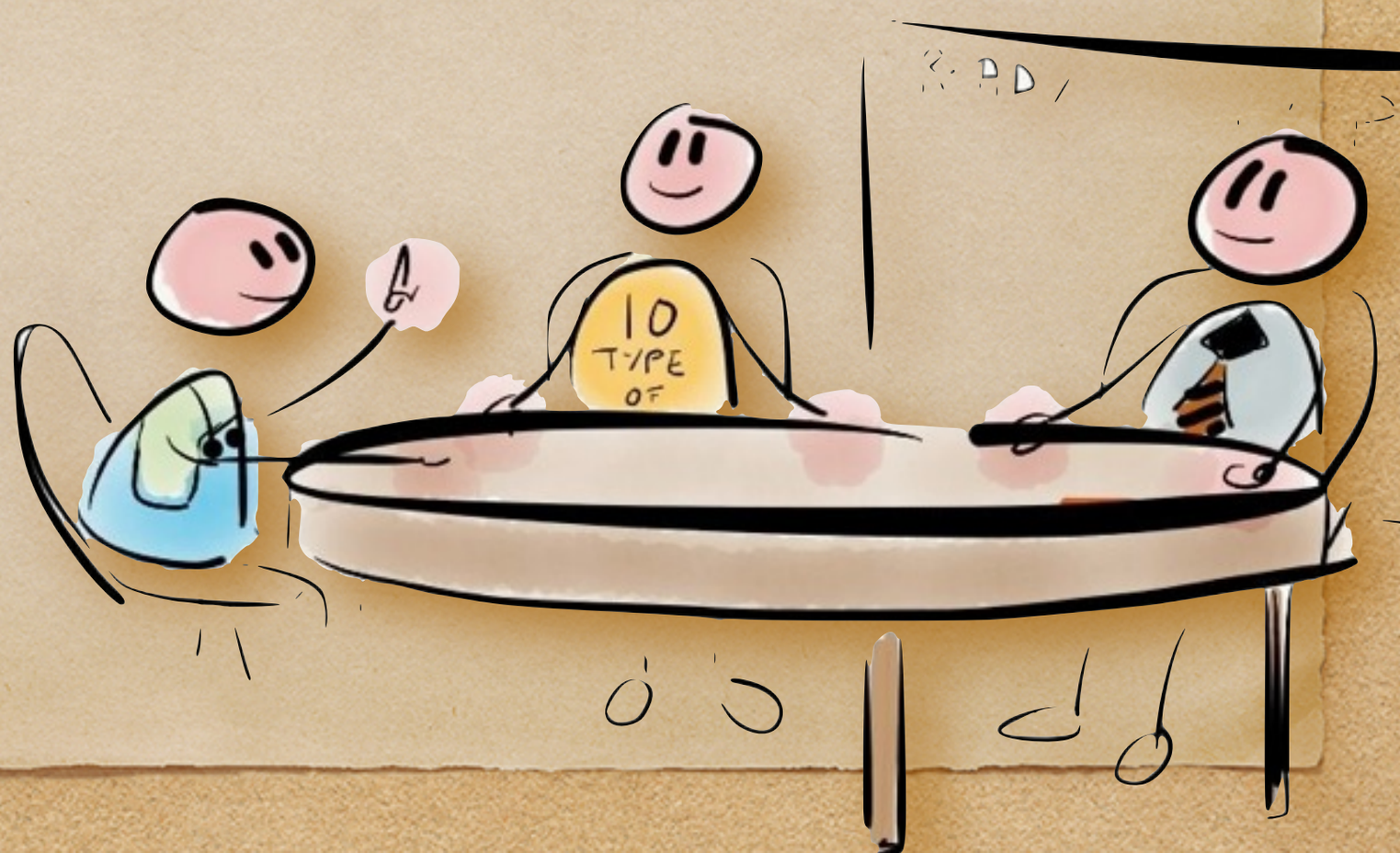
~~How!~~



Why? What?

Goal?

~~How!~~



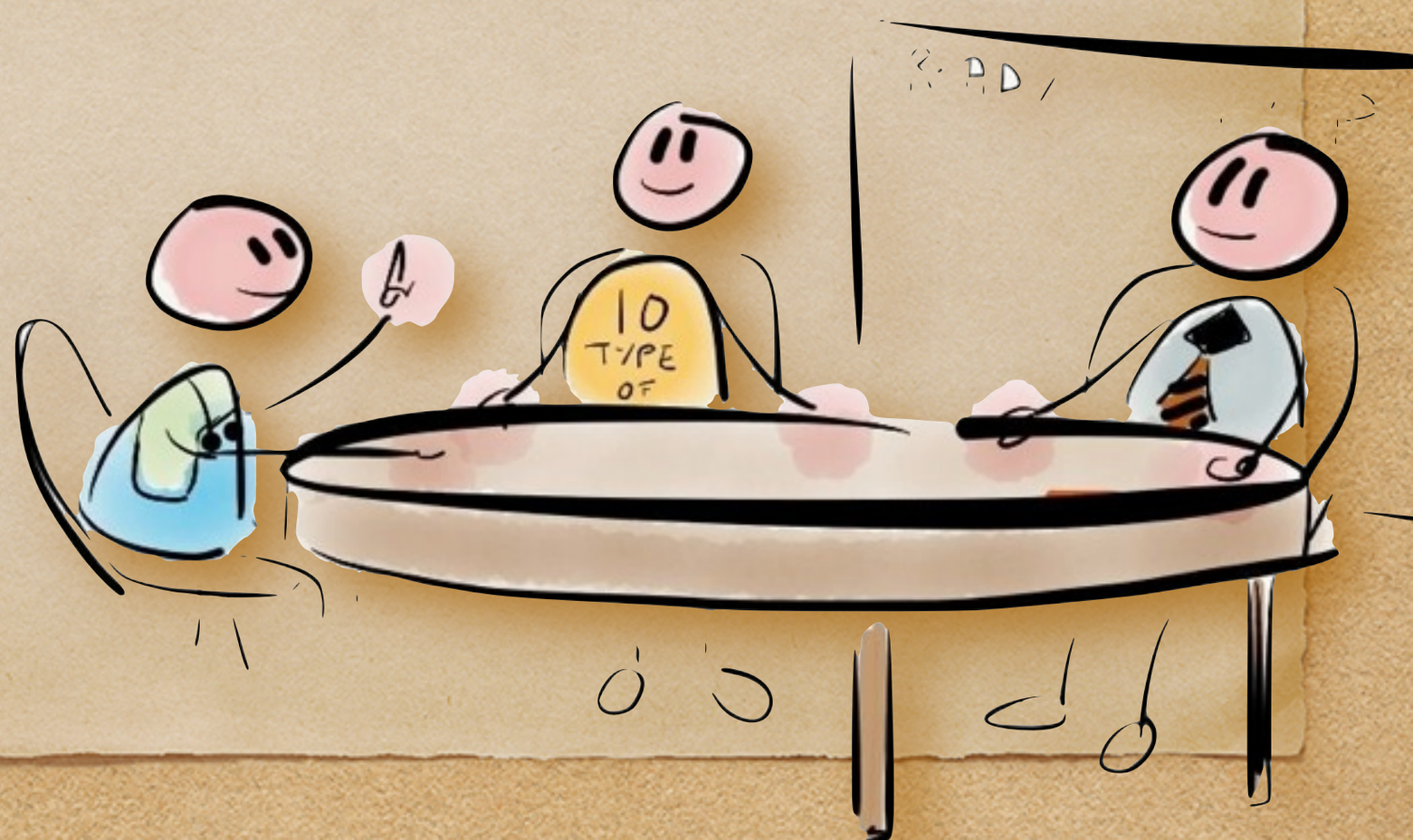
Intention

Why? What?

Goal?

Business Value?

~~How!~~



Feature: ...

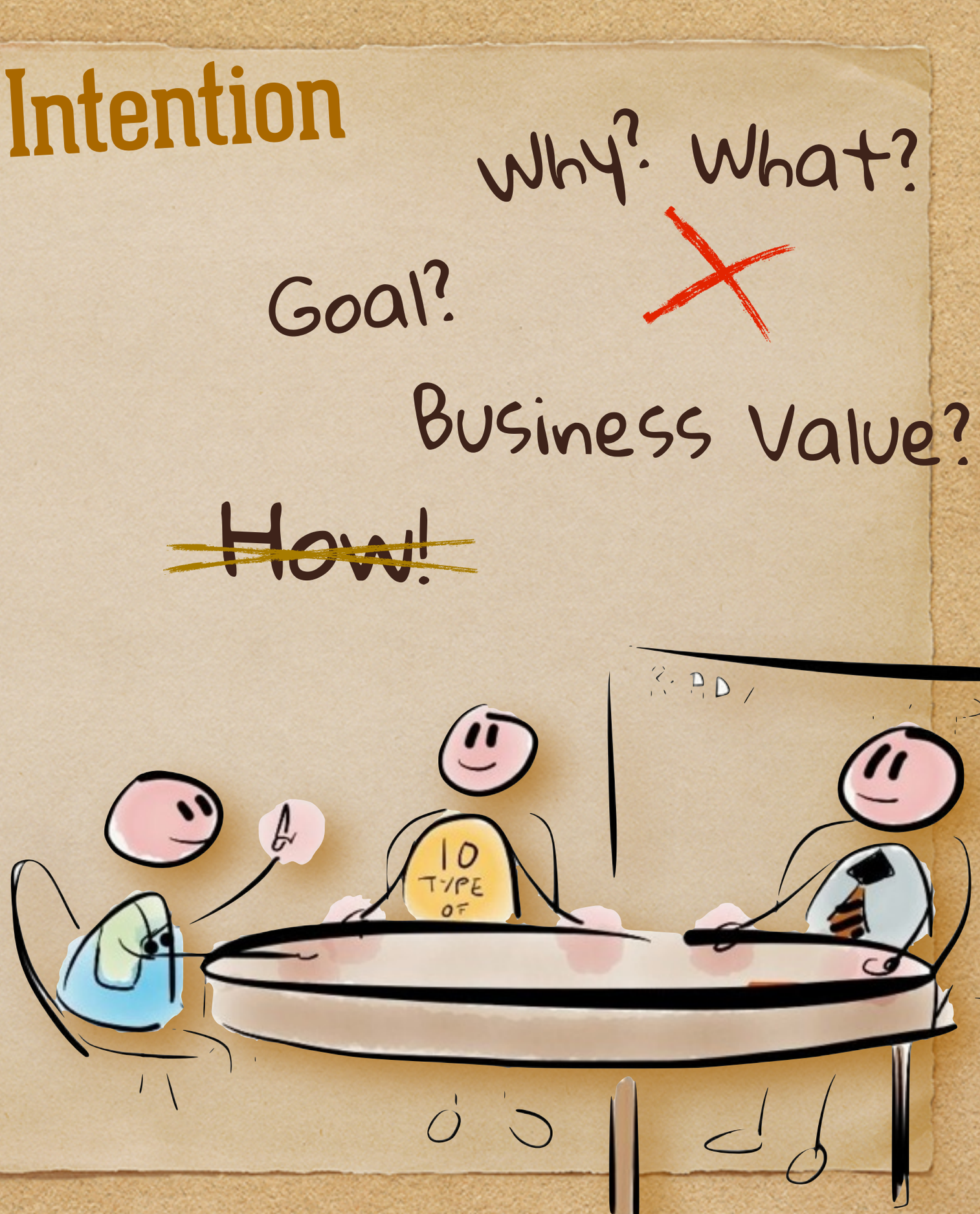
Intention

Why? What?

Goal?

Business Value?

~~How!~~



Feature: ...

Intention

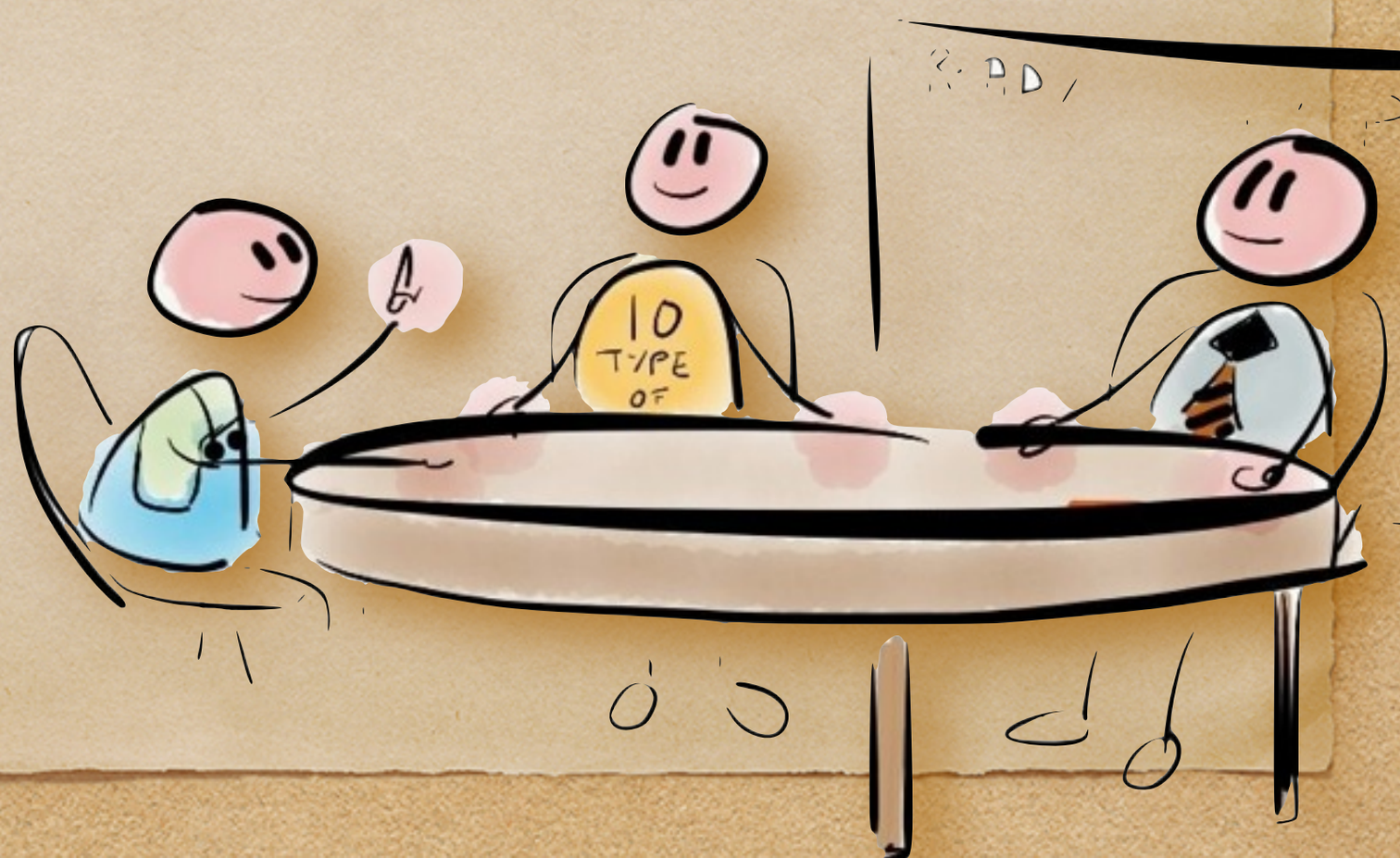
In order to <achieve the vision>

Why? What?

Goal?

Business Value?

~~How!~~



Feature: ...

Intention

In order to <achieve the vision>

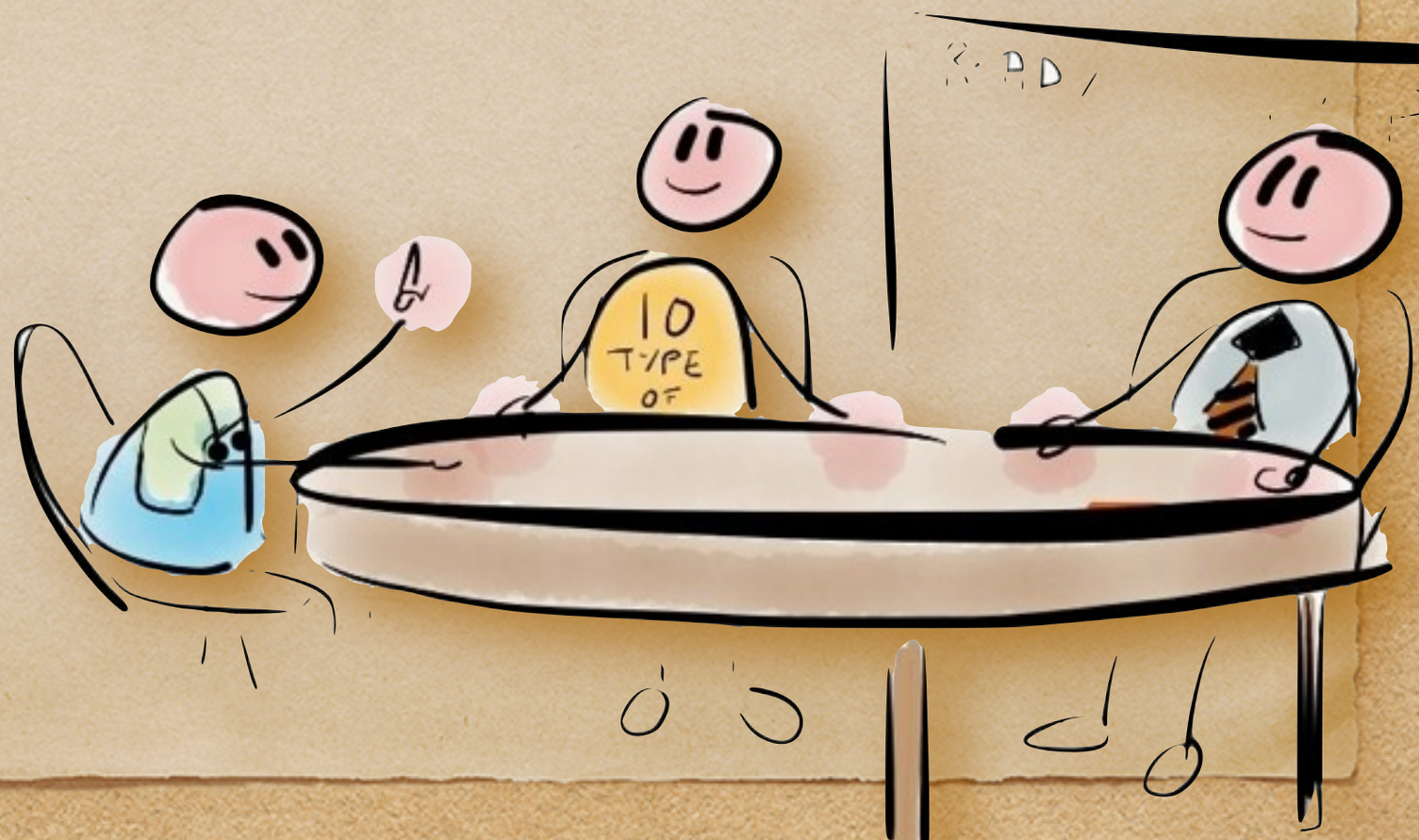
As a <stakeholder>

Why? What?

Goal?

Business Value?

~~How!~~



Feature: ...

Intention

In order to <achieve the vision>

As a <stakeholder>

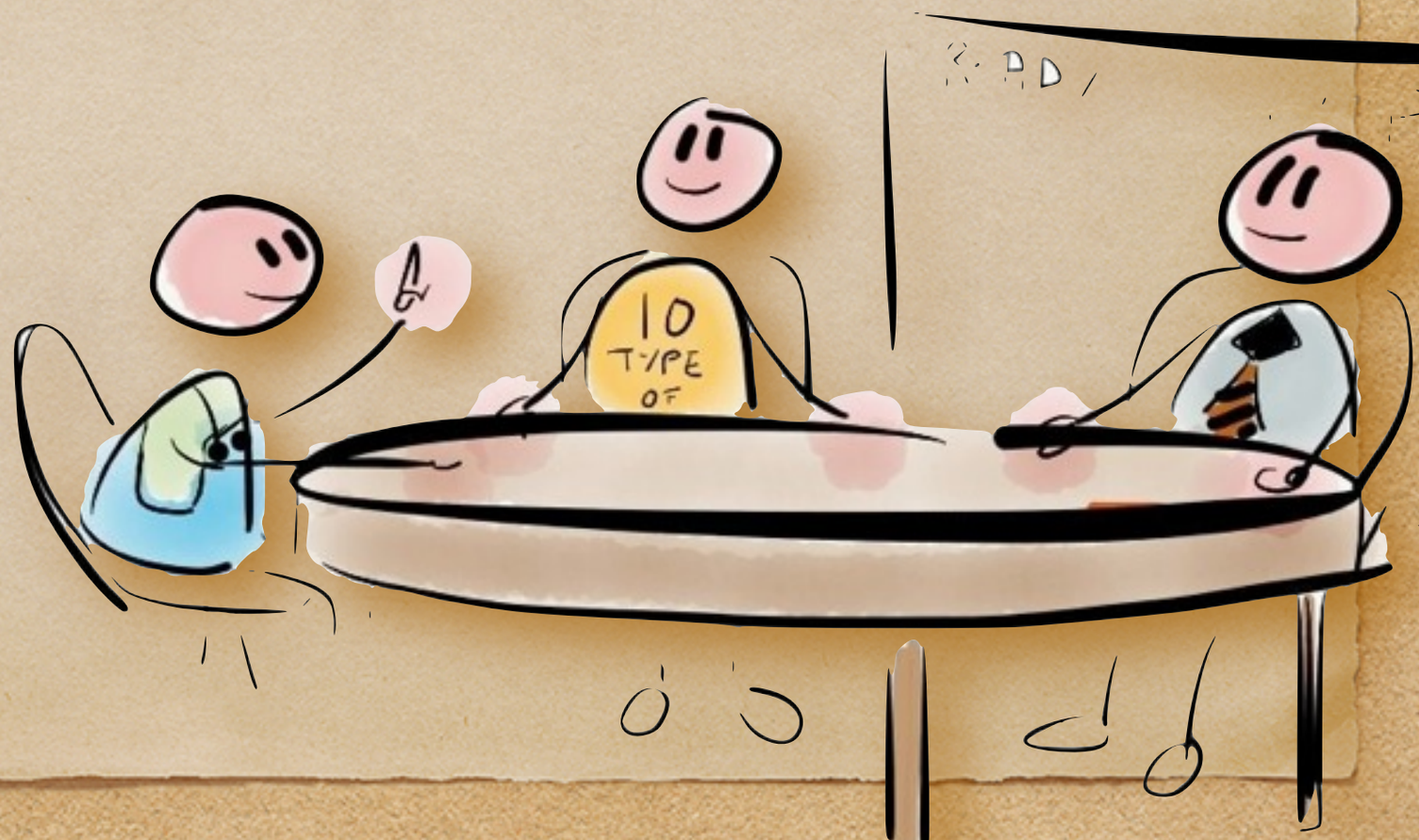
I want <value>

Why? What?

Goal?

Business Value?

~~How!~~



Feature: ...

Intention

In order to <achieve the vision>
As a <stakeholder>
I want <value>

Why? What?

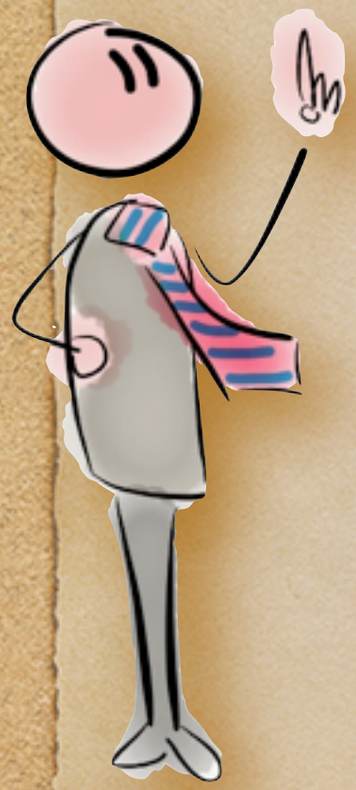
Goal?

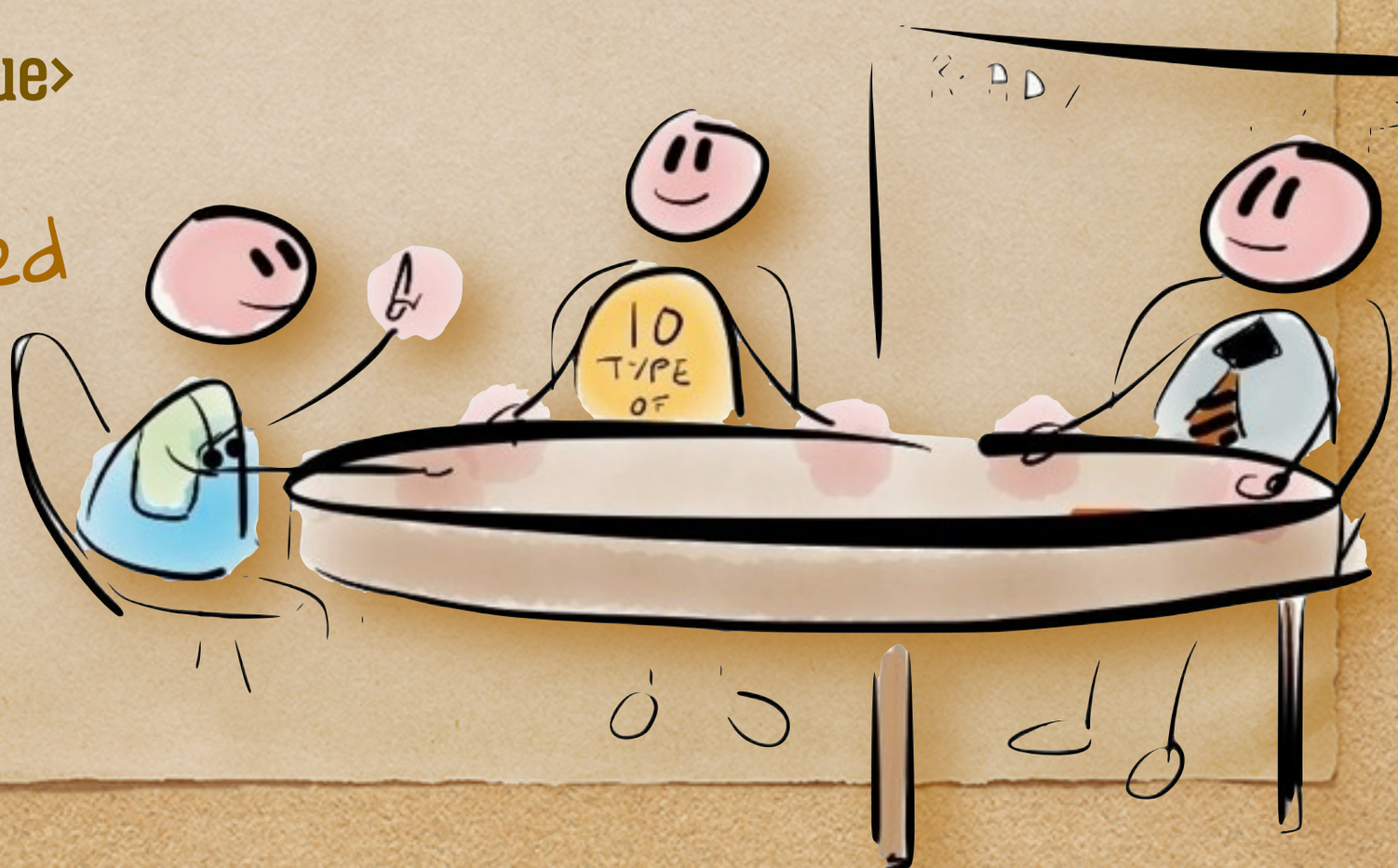
Business Value?

~~How!~~

As a <role>
I want <goal>
So that <value>

User focused





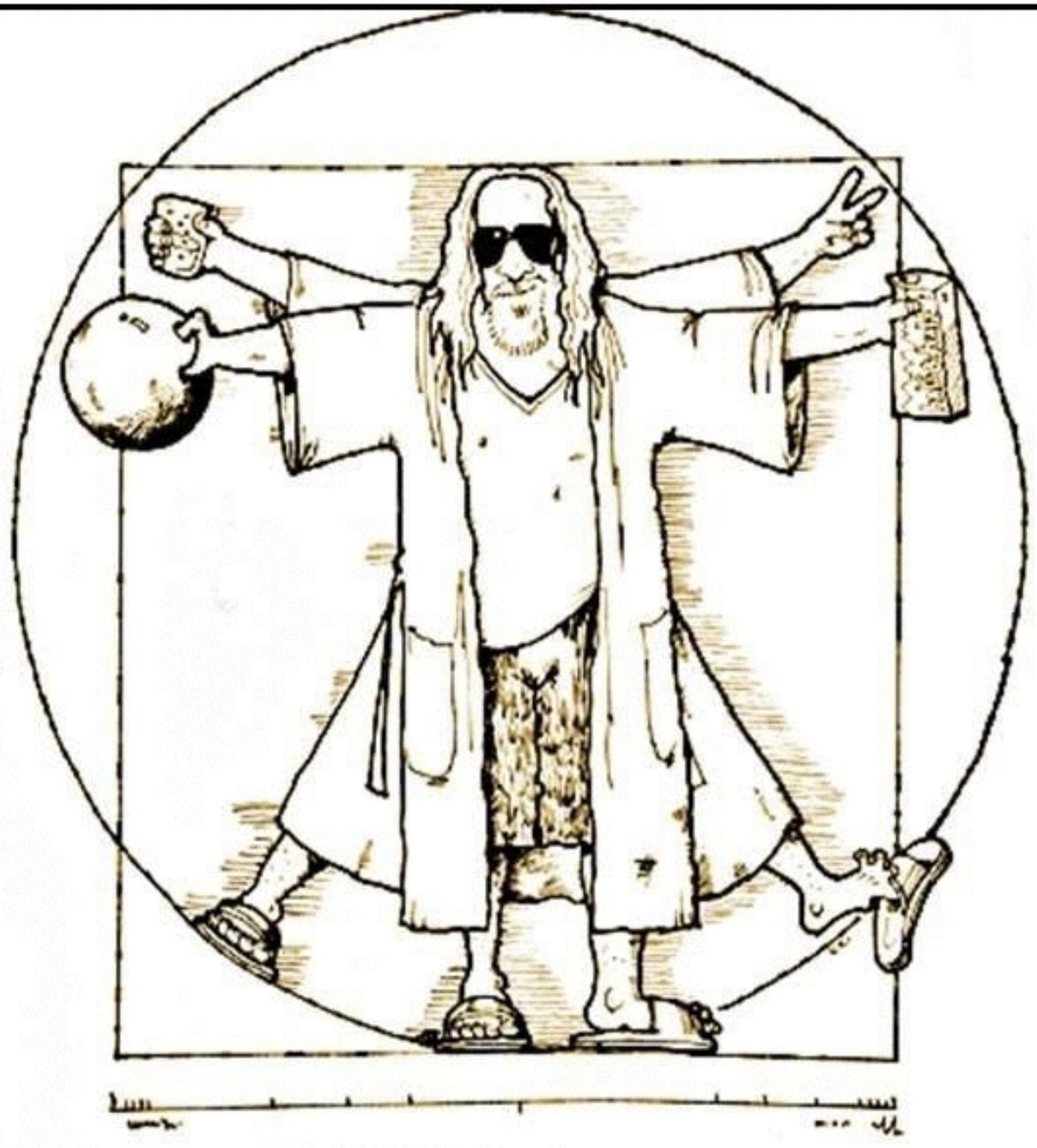
No Narrative

No Business Value

Unnecessary Feature !

Dude's Law: Value = Why / How

$$V = \frac{W}{H}$$



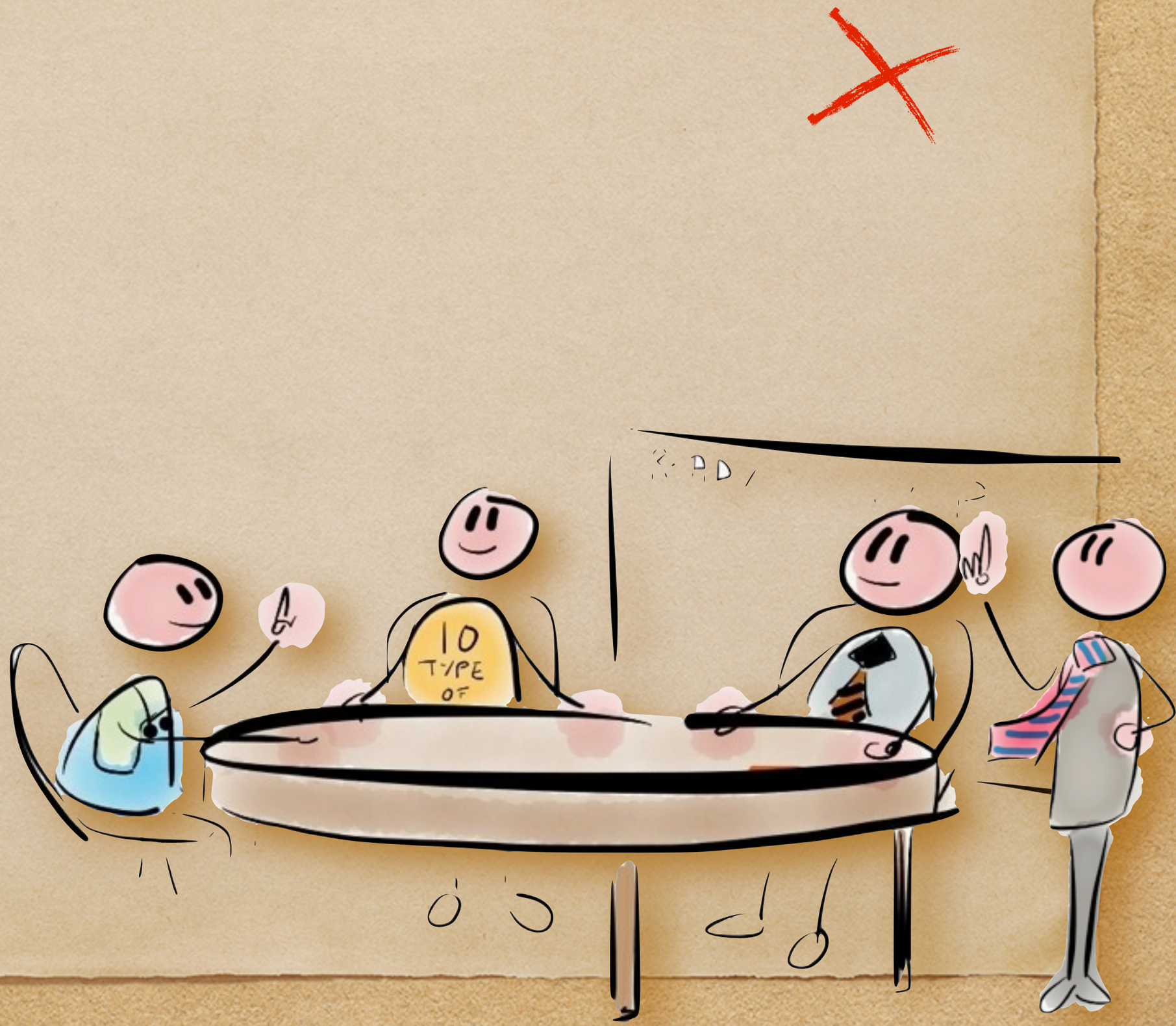
$$V = \frac{W}{H}$$

Feature: ...

In order to...

As a...

I want to...

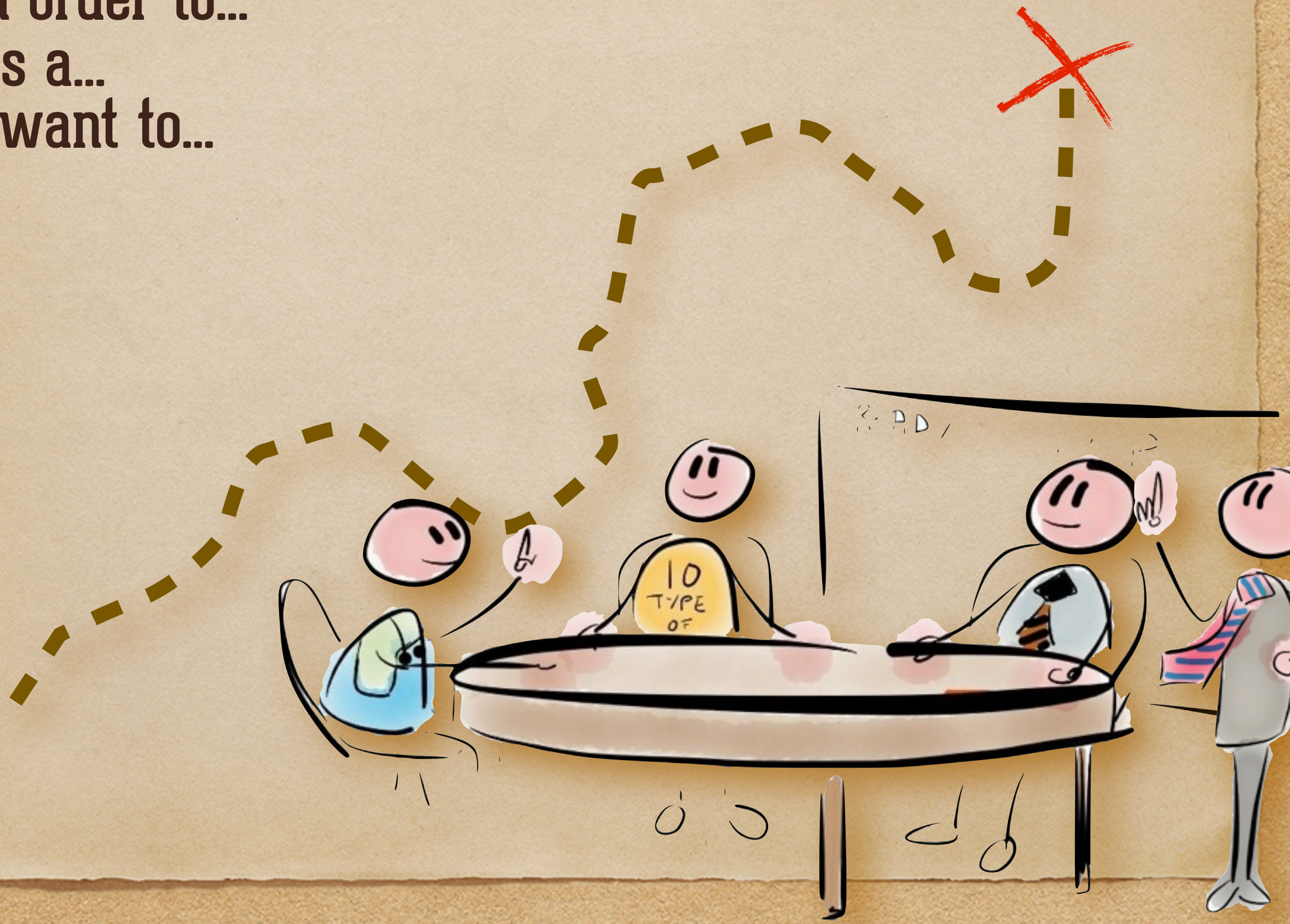


Feature: ...

In order to...

As a...

I want to...

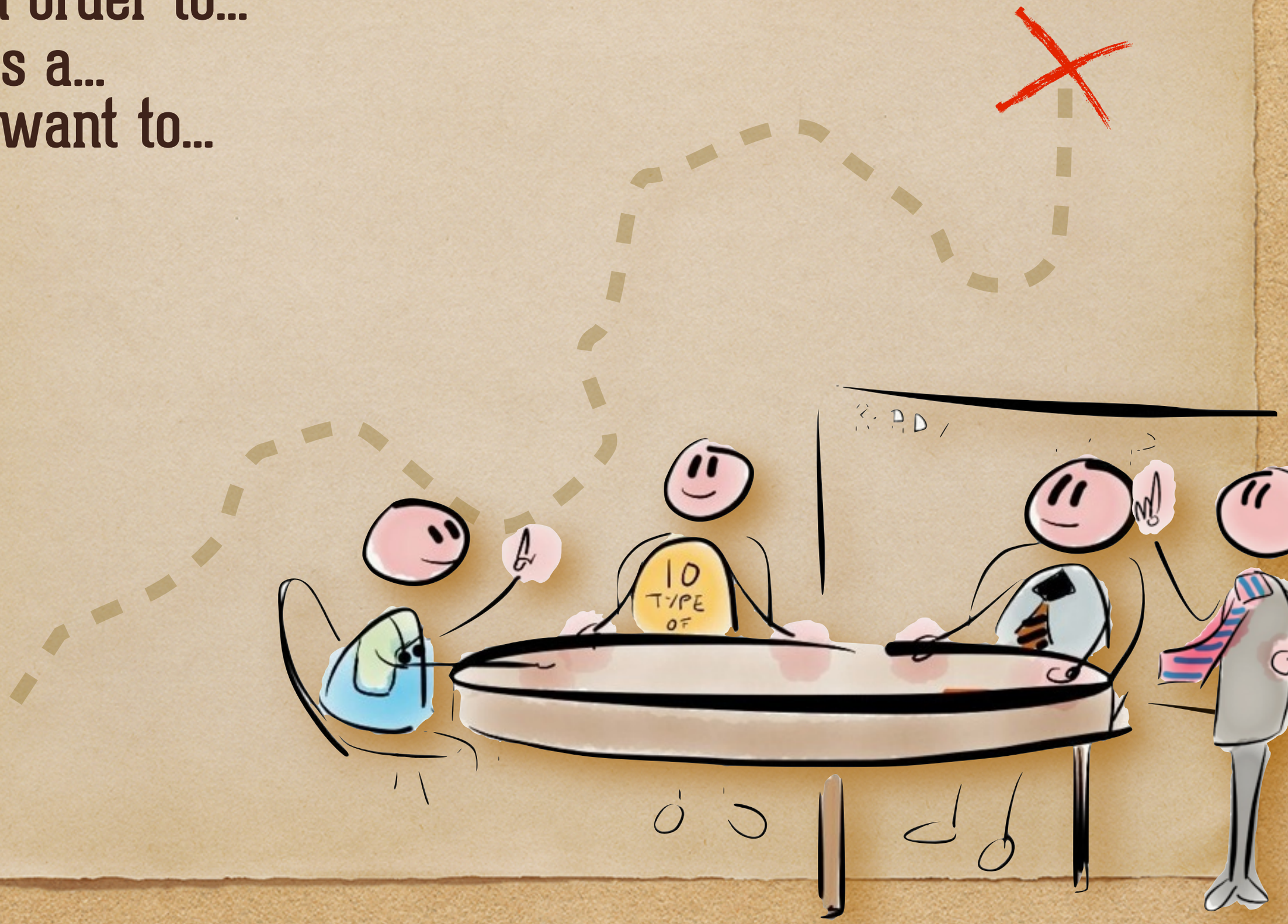


Feature: ...

In order to...

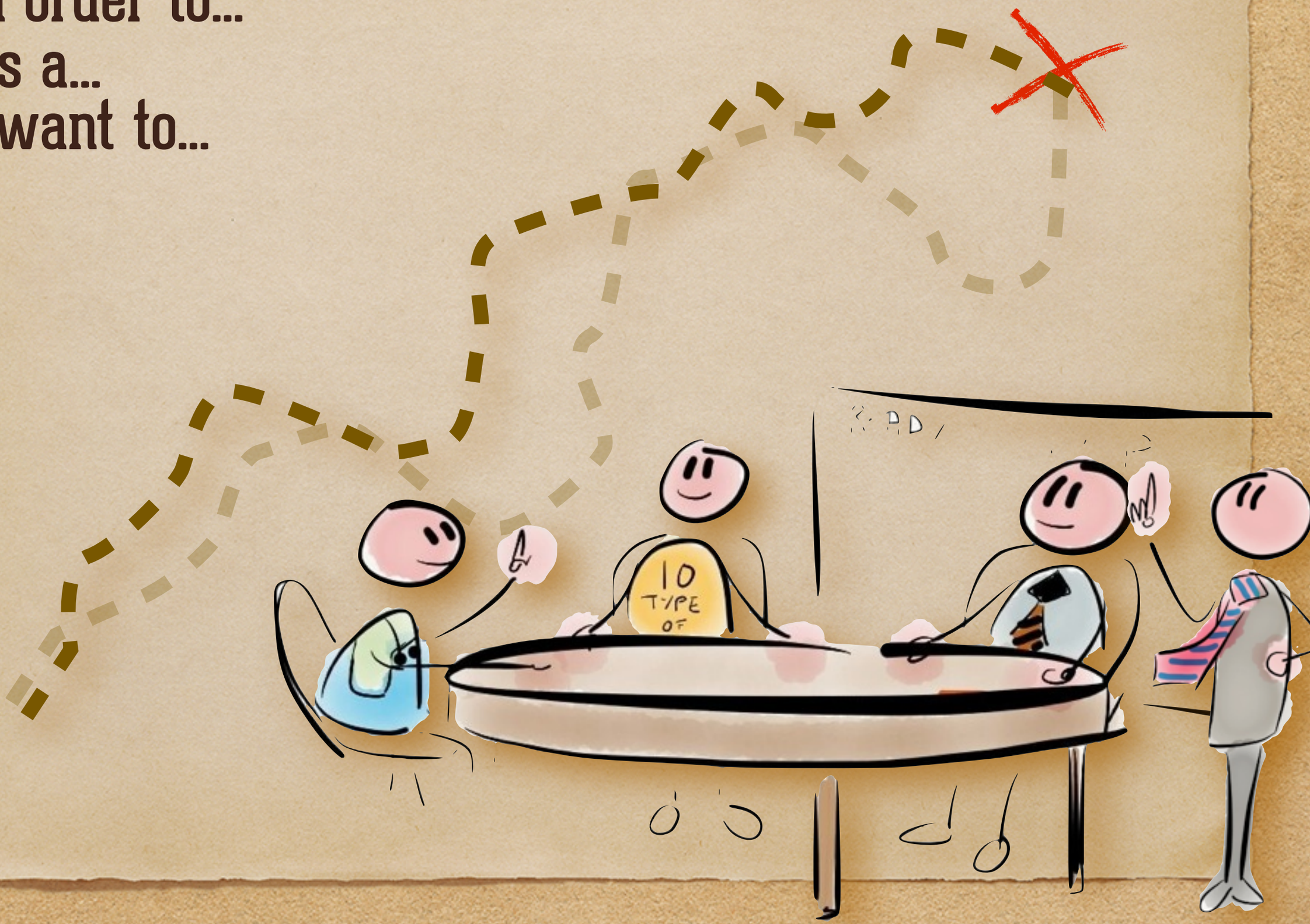
As a...

I want to...



Feature: ...

In order to...
As a...
I want to...



Feature: ...

In order to...

As a...

I want to...

Feature: ...

In order to...
As a...
I want to...

Scenario: ...

Feature: ...

In order to...
As a...
I want to...

Scenario: ...

Given <a context>

Feature: ...

In order to...

As a...

I want to...

Scenario: ...

Given <a context>

When <an event happens>

Feature: ...

In order to...
As a...
I want to...

Scenario: ...

Given <a context>
When <an event happens>
Then <an outcome **should** occur>

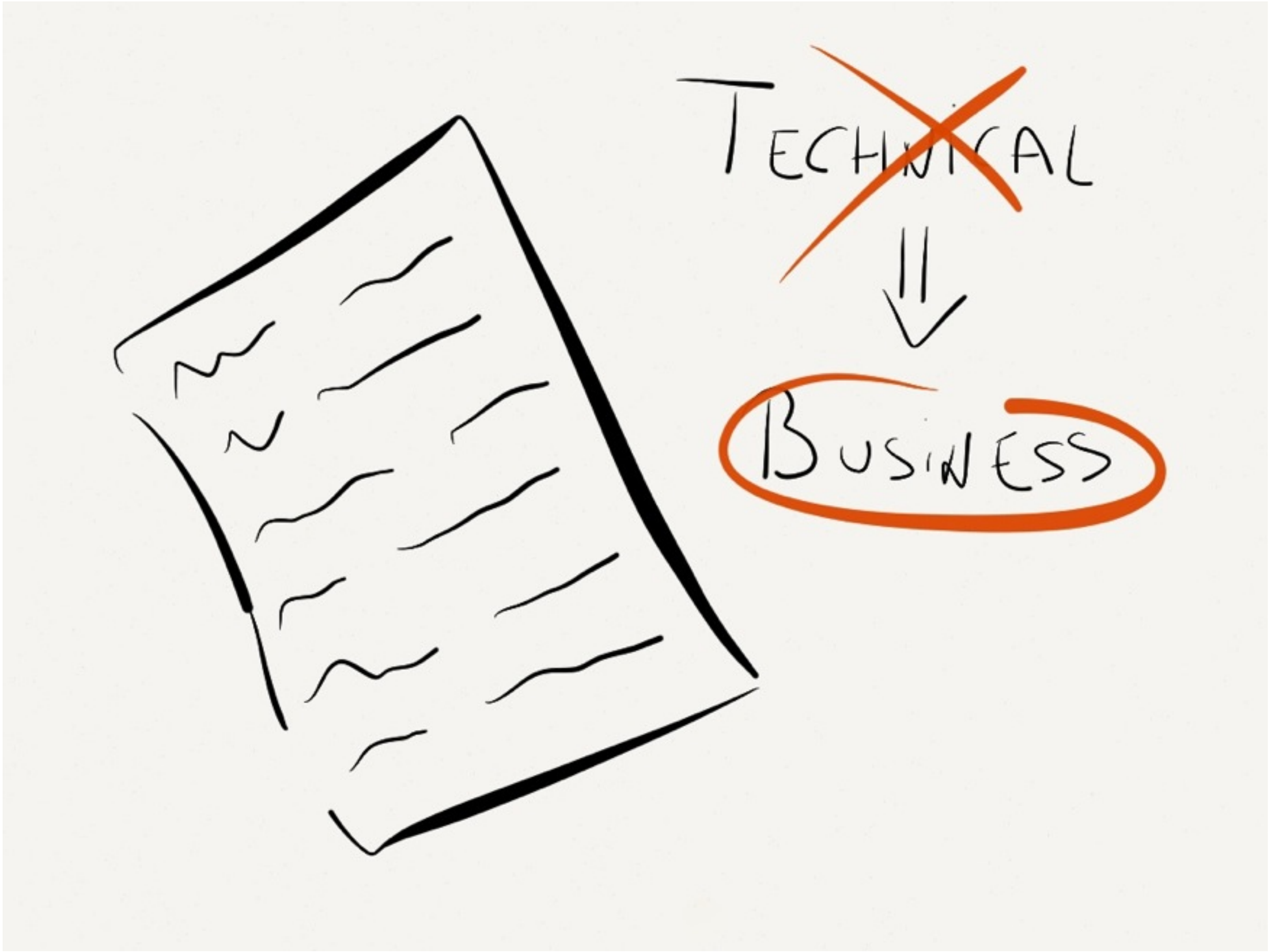
BDD uses examples to illustrate behavior

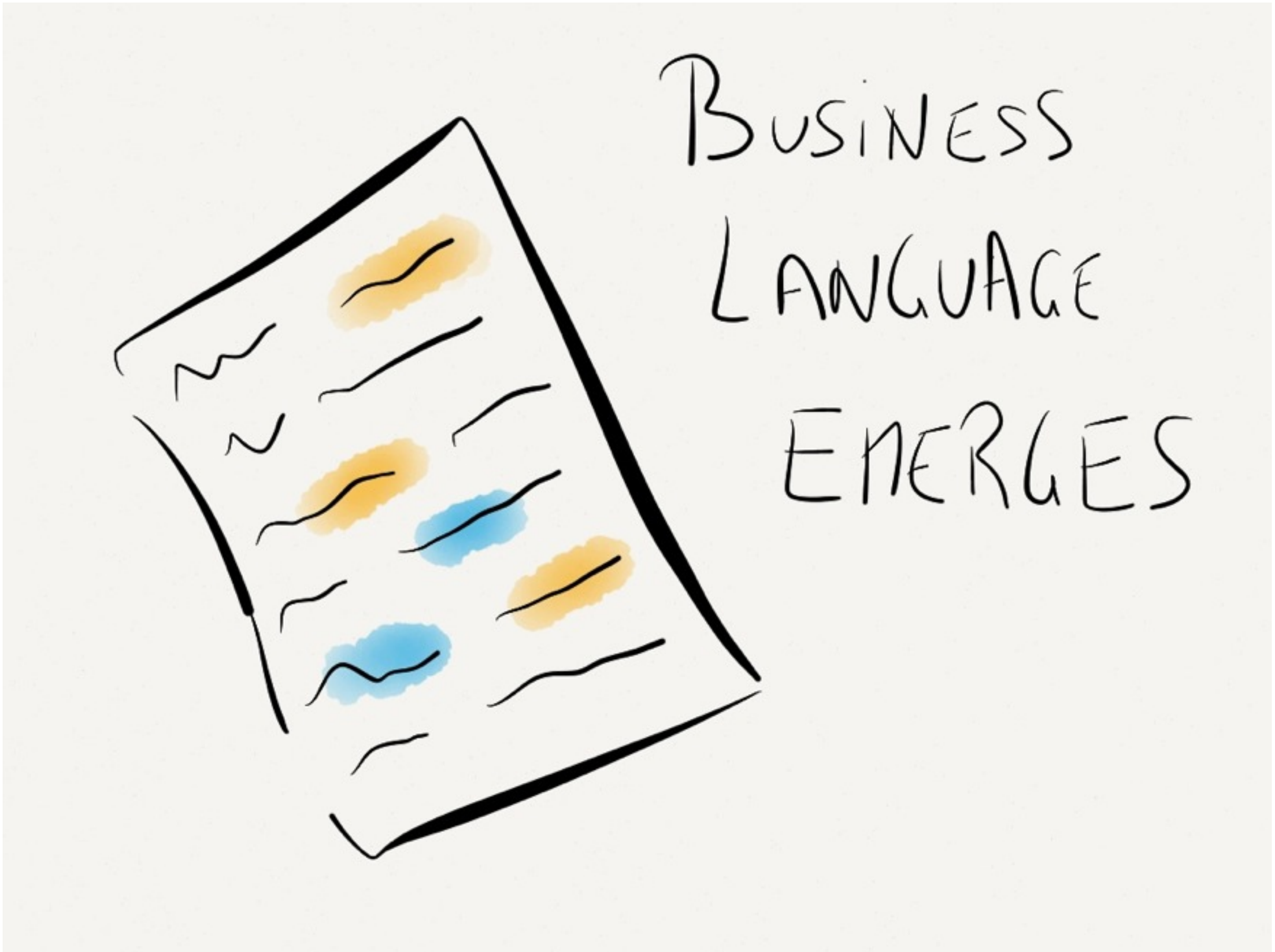
In order to...
As a...
I want to...

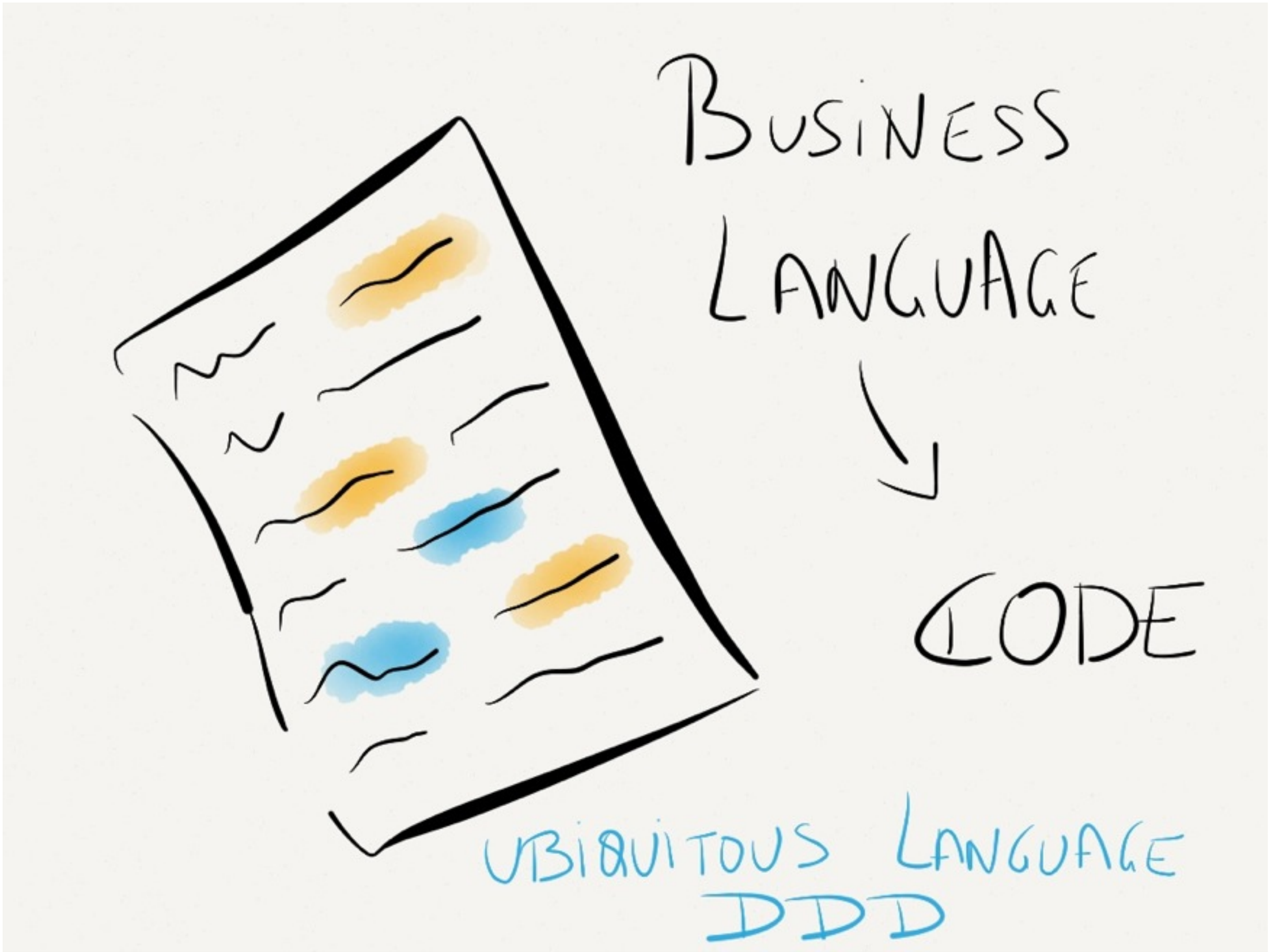
Scenario: ...
Given <a context>
When <an event happens>
Then <an outcome **should** occur>

SCENARIO

TEXT FILE


TECHNICAL
BUSINESS


BUSINESS
LANGUAGE
EMERGES

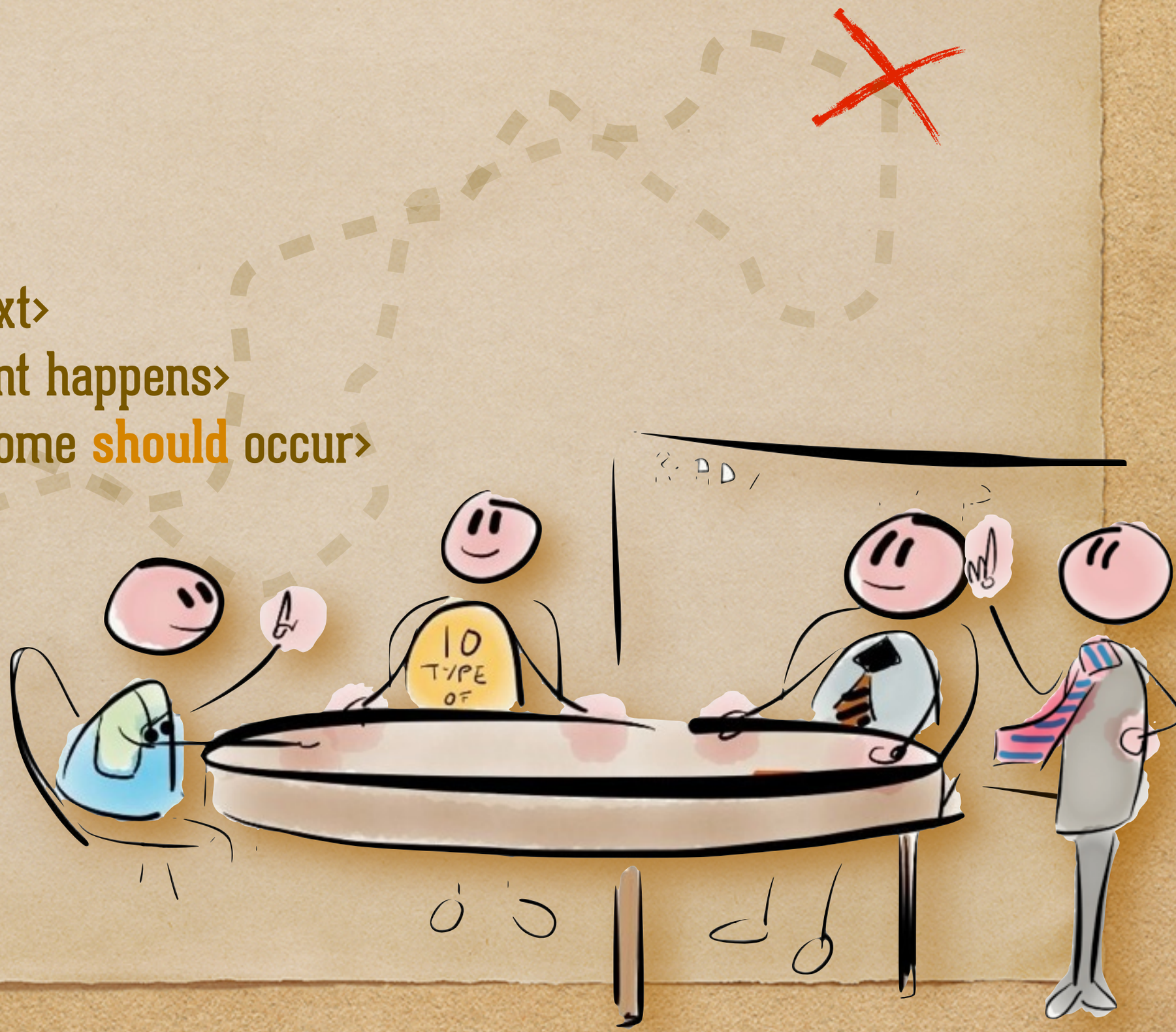

BUSINESS
LANGUAGE
CODE
UBIQUITOUS LANGUAGE
DDD

Feature: ...

In order to...
As a...
I want to...

Scenario: ...

Given <a context>
When <an event happens>
Then <an outcome **should** occur>



Feature: ...

In order to...
As a...
I want to...

Scenario: ...

Given <a context>
When <an event happens>
Then <an outcome should occur>

Examples help discover things early

In order to...
As a...
I want to...

Scenario: ...

Given <a context>
When <an event happens>
Then <an outcome should occur>

One may discover that one doesn't know

but others do!

Feature: ...

In order to...

As a...

I want to...

Scenario: ...

Given <a context>

When <an event happens>

Then <an outcome should occur>

Concrete Examples

Feature: Account Holder withdraws cash

As an Account Holder

I want to withdraw cash from an ATM

So that I can get money when the bank is closed

Scenario: Account has sufficient funds

Given the account balance is 100€

And the card is valid

And the machine contains enough money

When the Account Holder requests 20€

Then the ATM should dispense 20€

And the account balance should be 80€

And the card should be returned

Scenario: The ATM has insufficient funds

Given the account balance is 100€

And the card is valid

And the machine contains only 20€

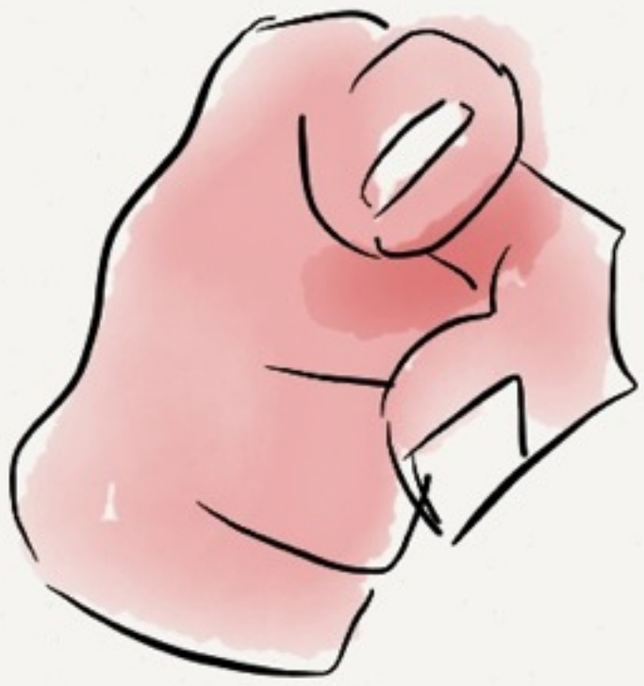
When the Account Holder requests 30€

Then the ATM should say it has insufficient funds

And the account balance should be 100€

And the card should be returned

Your Turn



Scenario: Account has insufficient funds

Given the account balance is 10€

And the card is valid

And the machine contains enough money

When the Account Holder requests 30€

Then the ATM should not dispense any money

And the ATM should say there are insufficient funds

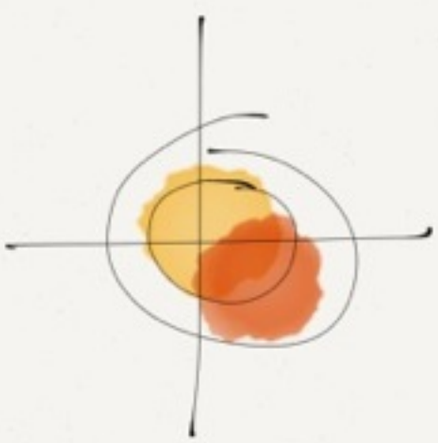
And the account balance should still be 10€

And the card should be returned

IMPLICIT

VS

EXPLICIT



Focus on the behavior described

Scenario: Wrong PIN

Given the card is valid

And its PIN number is "0000"

When the Account Holder enters "1234"

Then the ATM should say the PIN number is wrong

Scenario: Wrong PIN three times

Given the card is valid

And its PIN number is "0000"

When the Account Holder enters "1234"

And the Account Holder enters "4321"

And the Account Holder enters "2341"

Then the ATM should retain the card

And the ATM should say the card has been retained

Scenario: Card has been disabled

Given the card is disabled

When the Account Holder requests 30€

Then the ATM should retain the card

And the ATM should say the card has been retained

Scenario: Card has been disabled

Given the card is disabled

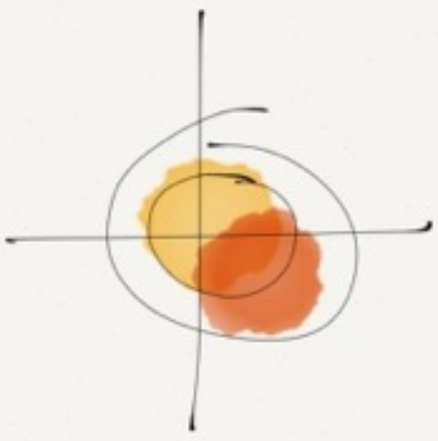
When the Account Holder requests 30€

Then the ATM should retain the card

And the ATM should say the card has been retained

```
Feature: Account Holder withdraws cash from an ATM

  In the following scenario, ATM will stands for Automatic Teller
  Machine in other word a “Cash machine”. Unless specified, one will
  assume that:
  * Account has enough money on its account,
  * The card is valid (not blocked, not expired, …)
  * The ATM has all bills in an infinite quantity.

  As an Account Holder
  In order to get money at any time, even when the bank is closed
  I want to withdraw cash from an ATM

Scenario: Account has sufficient funds
  Given the account balance is 100€
   When the Account Holder requests 20€
   Then the ATM should dispense 20€
    And the account balance should be 80€
    And the card should be returned

Scenario: The ATM has insufficient funds
  Given the machine contains only 20€
   When the Account Holder requests 30€
   Then the ATM should say it has insufficient funds
    And the account balance should remain unchanged
    And the card should be returned
```

Declarative vs Imperative



Focus on the behavior described

```
Feature:
  As a User
  I want to create a new user registration to the site
  So that I can share my profile with the community

  Scenario: successful submission

  Given I'm on the user creation page
   When I fill in Name with ‘Jean-Paul’
    And select Gender as ‘Male’
    And fill in Username with 'jp'
    And fill in Password with 's3cr3t'
    And select in Occupation with 'Private Job'
    And check Terms and Conditions
    And click the Create button
   Then I should see the notice 'Thank you for your submission!'
    And the page should include the user name, gender, username,
occupation
```

```
Feature:
  As a User
  I want to create a new user registration to the site
  So that I can share my profile with the community

  Scenario: successful submission

  Given I'm on the user creation page
   When I add a new user
    And fill in Name with ‘Jean-Paul’
    And select Gender as ‘Male’
    And fill in Username with 'jp'
    And fill in Password with 's3cr3t'
    And select in Occupation with 'Private Job'
    And check Terms and Conditions
    And click the Create button
   Then I should see the page for my newly created profile
    And the notice 'Thank you for your submission!'
    And the page should include the submitted gender, username,
occupation
```


BDD - CLINIC

Why this feature ?

Feature: Interpolate

In order to interpolate values

As an Trader

I want to interpolate values in a range of Market data

Why this feature ?

Feature: Interpolate

In order to interpolate values

As an Trader

I want to interpolate values in a range of Market data

Why this feature ?

Feature: Linear Interpolation

In order to fill the gaps and provide a value for any maturity

As a trader responsible for market-marking

I want to interpolate linearly values within a range of points

And **I want** a flat extrapolation outside of the range of points

What about this scenario

Scenario: Change the negotiation price from positive to negative => soulte cashflow appears and premium cashflow is modified
Given an FUNKY_EXOTIC
And deal way is sell
And deal nature is TOMATO
And trade value date is 2012/07/01
And nominal is 100 JPY
And negotiation price is 0.20 JPY
When I validate the deal
Then there are 1 Price cashflows
And there are 0 fee cashflows
When I change the negotiation price to -0.3 JPY
And I validate the deal
Then there are 1 Price cashflows
And there are 1 fee cashflows
And the trade cashflow's payment date is 2012/07/01
And the trade cashflow's way is receive
And the trade cashflow's amount is 30 JPY
And the fee cashflow's payment date is 2012/07/01
And the fee cashflow's way is give
And the fee cashflow's amount is 60 JPY

What about this scenario

Scenario: Change the negotiation price from positive to negative => soulte cashflow appears and premium cashflow is modified
Given an FUNKY_EXOTIC
And deal way is sell
And deal nature is TOMATO
And trade value date is 2012/07/01
And nominal is 100 JPY
And negotiation price is 0.20 JPY
When I validate the deal
Then there are 1 Price cashflows
And there are 0 fee cashflows
When I change the negotiation price to -0.3 JPY ...
And I validate the deal
Then there are 1 Price cashflows
And there are 1 fee cashflows
And the trade cashflow's payment date is 2012/07/01
And the trade cashflow's way is receive
And the trade cashflow's amount is 30 JPY
And the fee cashflow's payment date is 2012/07/01
And the fee cashflow's way is give
And the fee cashflow's amount is 60 JPY

```
d = new Deal();
d.SetWay(Sell);
d.SetNature(Tomato);
d.SetValueDate(new Date(...));
d.SetNominal(100, JPY);
d.SetNegotiationPrice(0.20, JPY);
cf = d.GetCashFlows();
AssertThat(IsEqual(...);
```

What about this scenario

Scenario: Fee and Price cashflows when the negotiation price is set to a negative value

Given a sell for a nominal 100 JPY on FUNKY_EXOTIC TOMATO negotiation price 0.20 JPY
traded on 2012/07/01

When the middle officer validates the deal

Then the trade has one Price cashflow and no Fee cashflow

When the middle officer changes the negotiation price to -0.3 JPY

And the middle officer validates the deal

Then the trade has the following cashflows:

FlowType	Way	Amount	Currency	PaymentDate	Remarks
Price	Receive	30	JPY	2012/07/01	100*abs(-0.3)
Fee	Give	60	JPY	2012/07/01	100*2*abs(-0.3)

What about this scenario

Scenario: Fee and Price cashflows when the negotiation price is set to a negative value

Given a sell for a nominal 100 JPY on FUNKY_EXOTIC TOMATO negotiation price 0.20 JPY
traded on 2012/07/01

When the middle officer validates the deal

Then the trade has one Price cashflow and no Fee cashflow

When the middle officer changes the negotiation price to -0.3 JPY

And the middle officer validates the deal

Then the trade has the following cashflows:

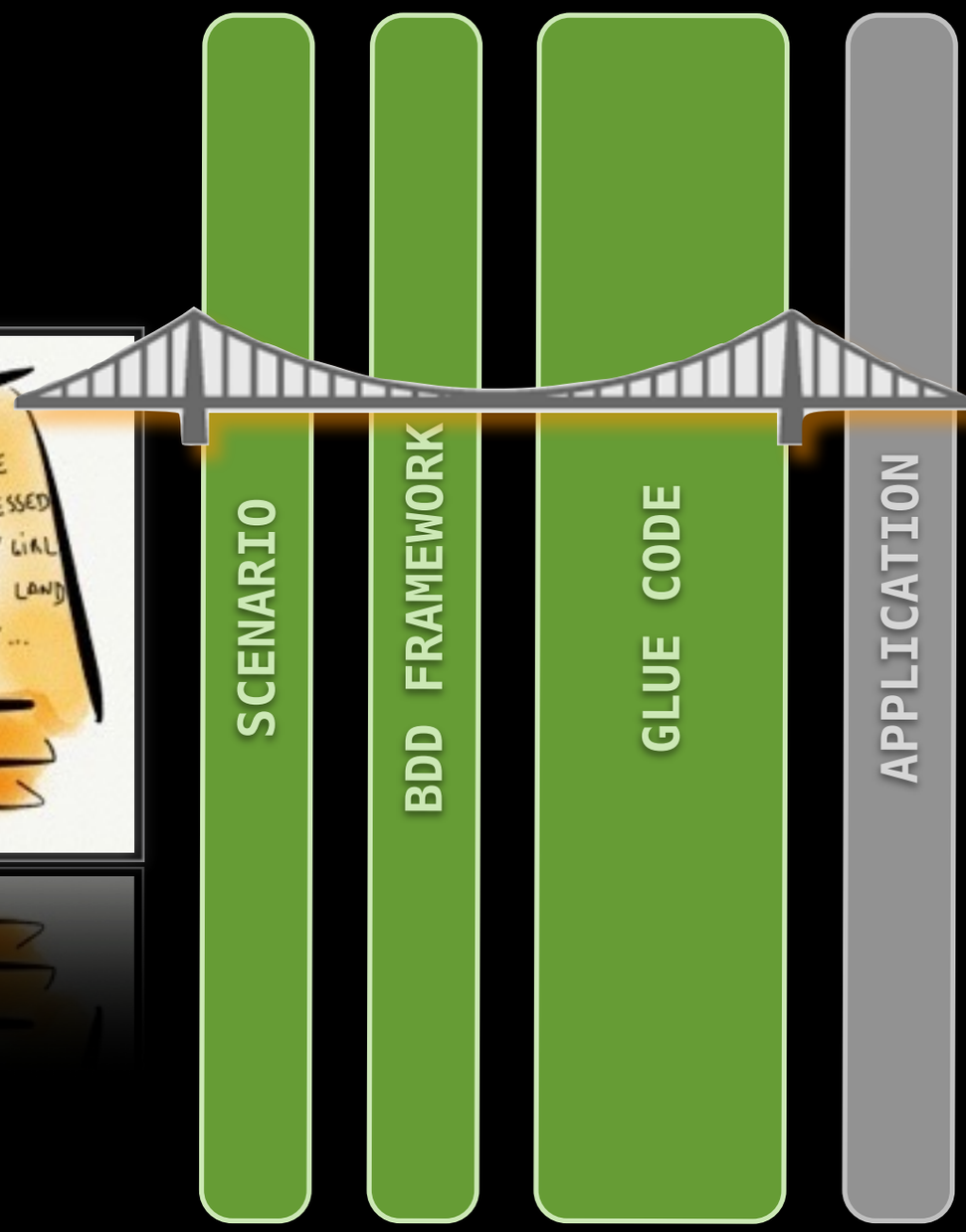
FlowType	Way	Amount	Currency	PaymentDate	Remarks
Price	Receive	30	JPY	2012/07/01	100*abs(-0.3)
Fee	Give	60	JPY	2012/07/01	100*2*abs(-0.3)

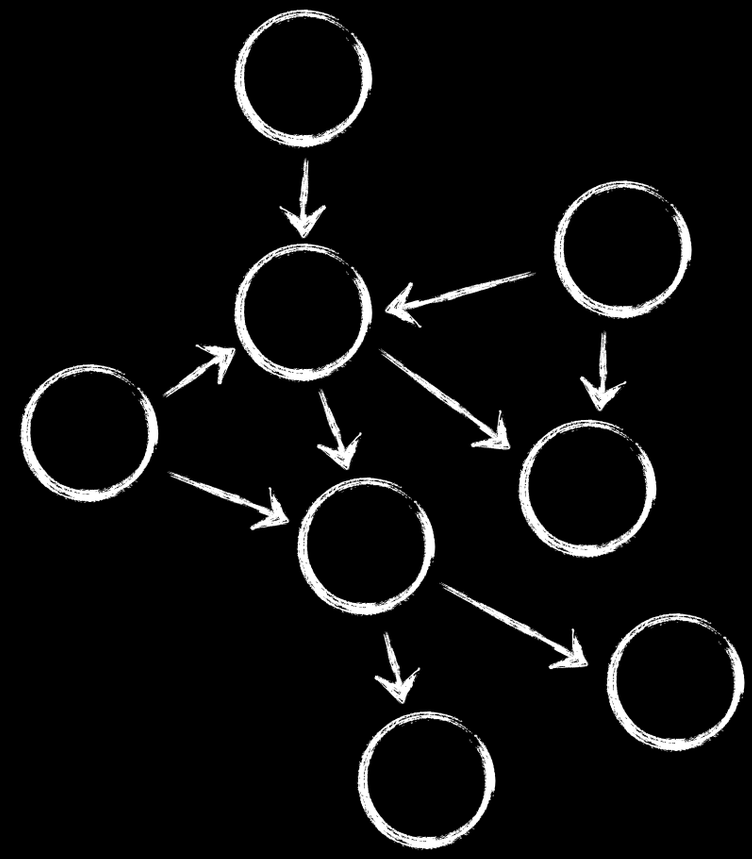
Communicate With the Business People !!!

AUTOMATION

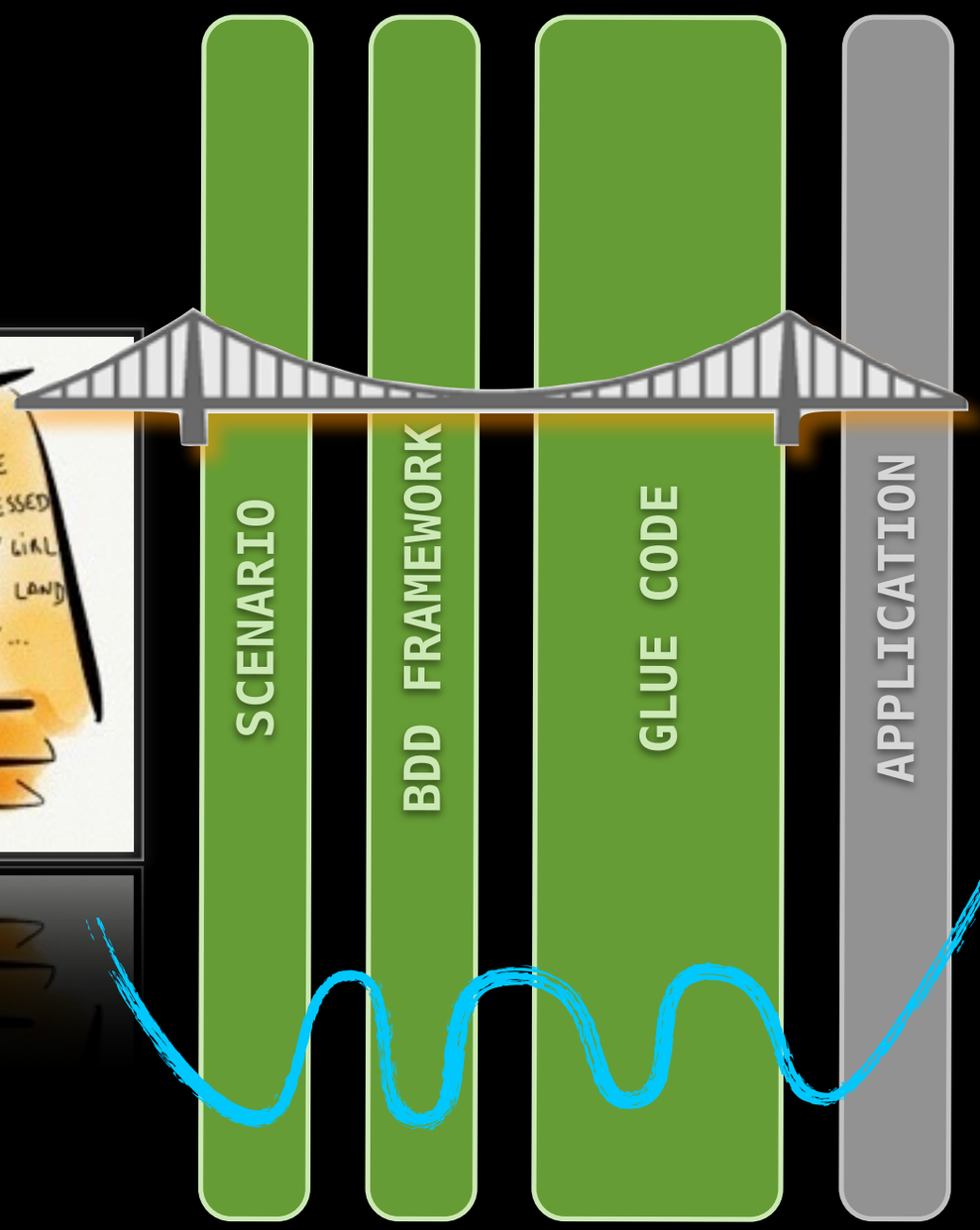
BRIDGE BETWEEN
SCENARIO & APP.

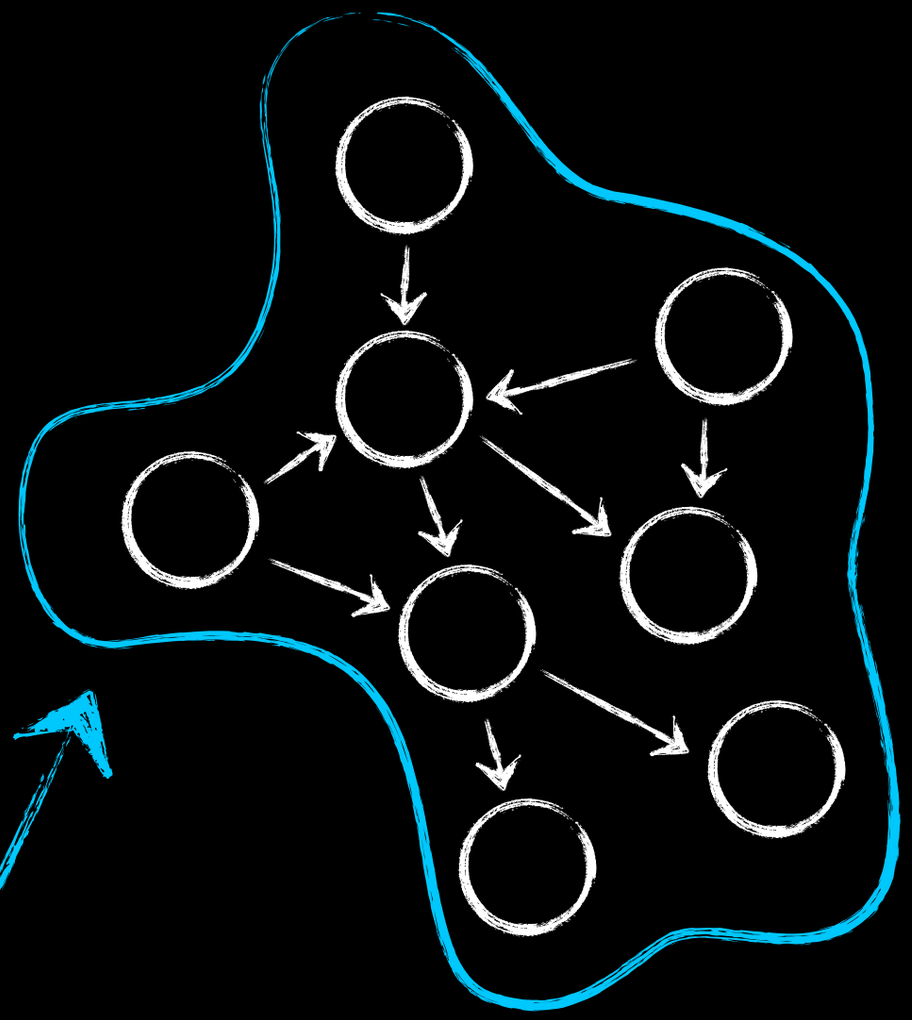

ONCE UPON A TIME IN A
KINGDOM FAR, FAR AWAY, THE
KING AND QUEEN WERE BLESSED
WITH A BEAUTIFUL BABY GIRL
AND THROUGHOUT THE LAND
EVERYONE WAS HAPPY...


SCENARIO
BDD FRAMEWORK
GLUE CODE
APPLICATION




ONCE UPON A TIME IN A
KINGDOM FAR, FAR AWAY, THE
KING AND QUEEN WERE BLESSED
WITH A BEAUTIFUL BABY GIRL
AND THROUGHOUT THE LAND
EVERYONE WAS HAPPY...

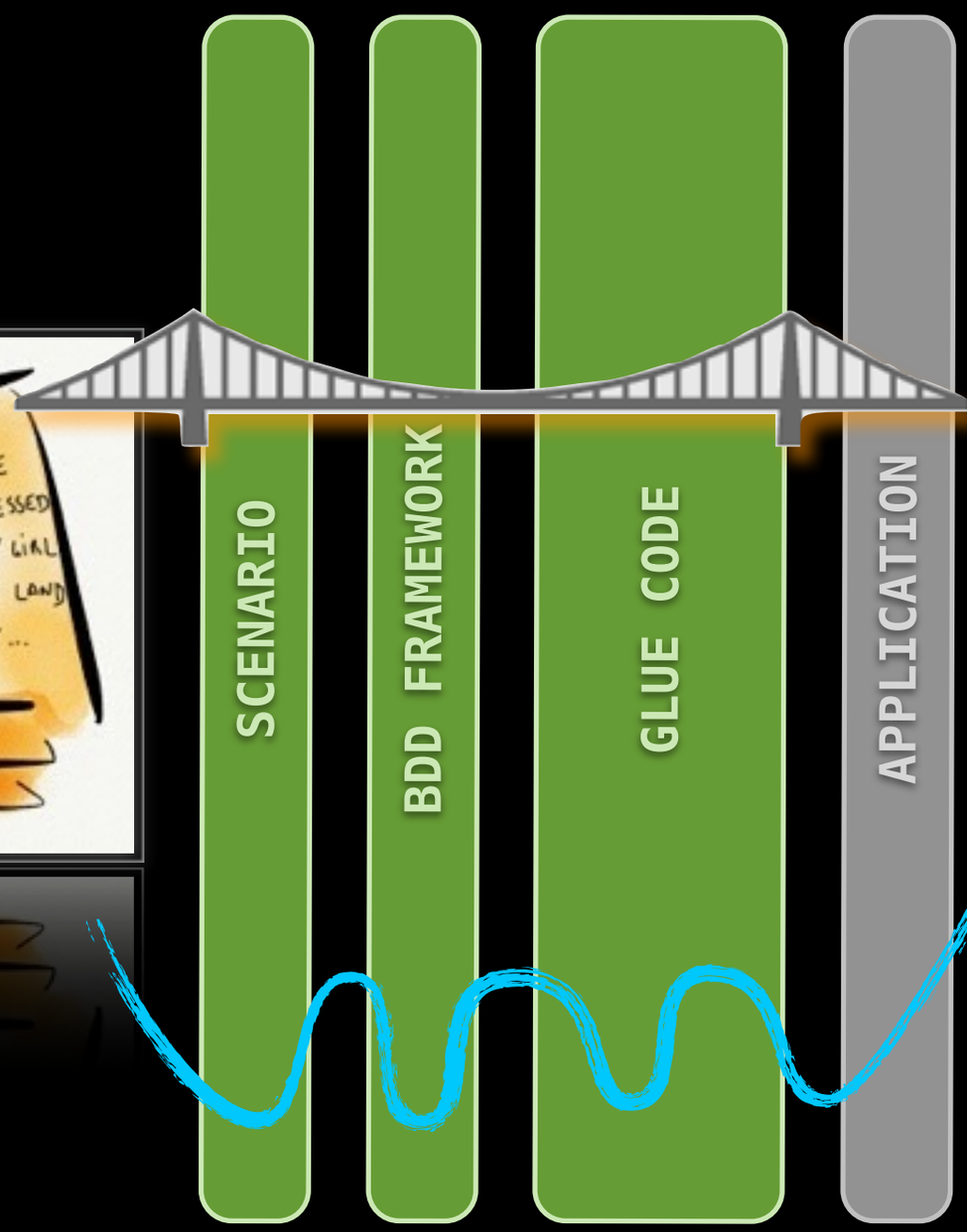

SCENARIO
BDD FRAMEWORK
GLUE CODE
APPLICATION

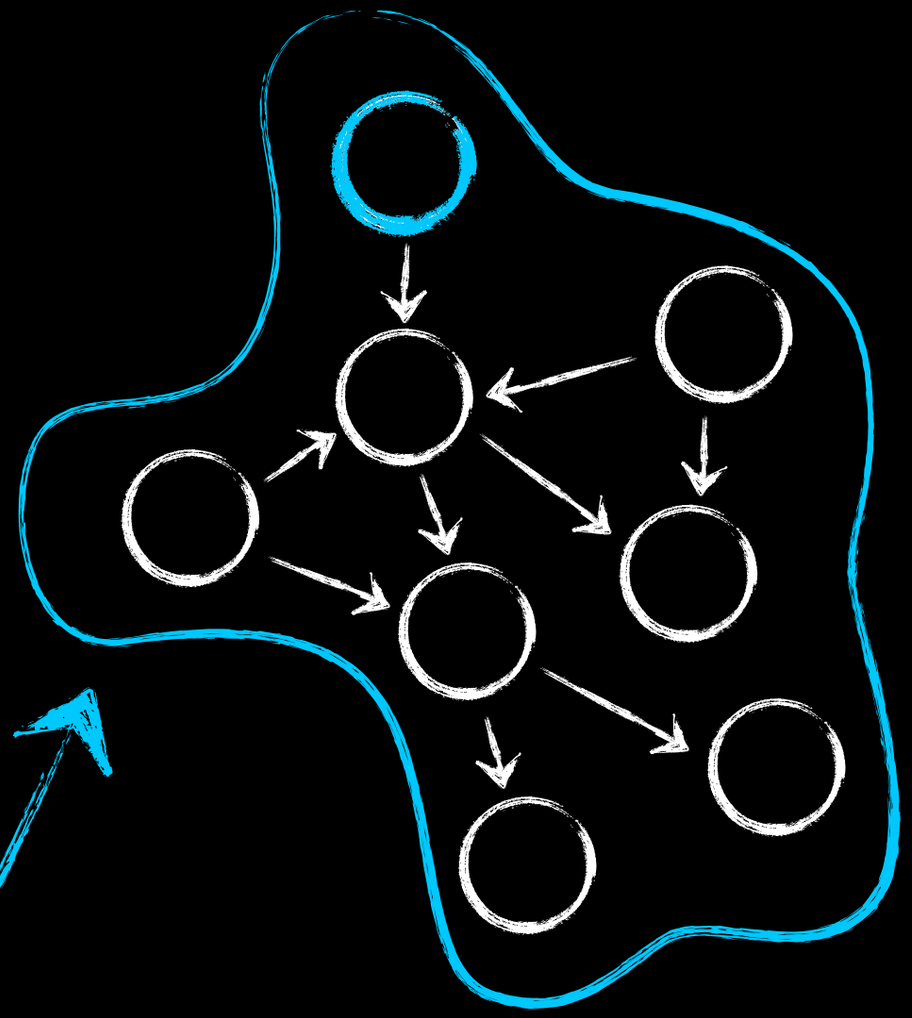


"End to End"

UI

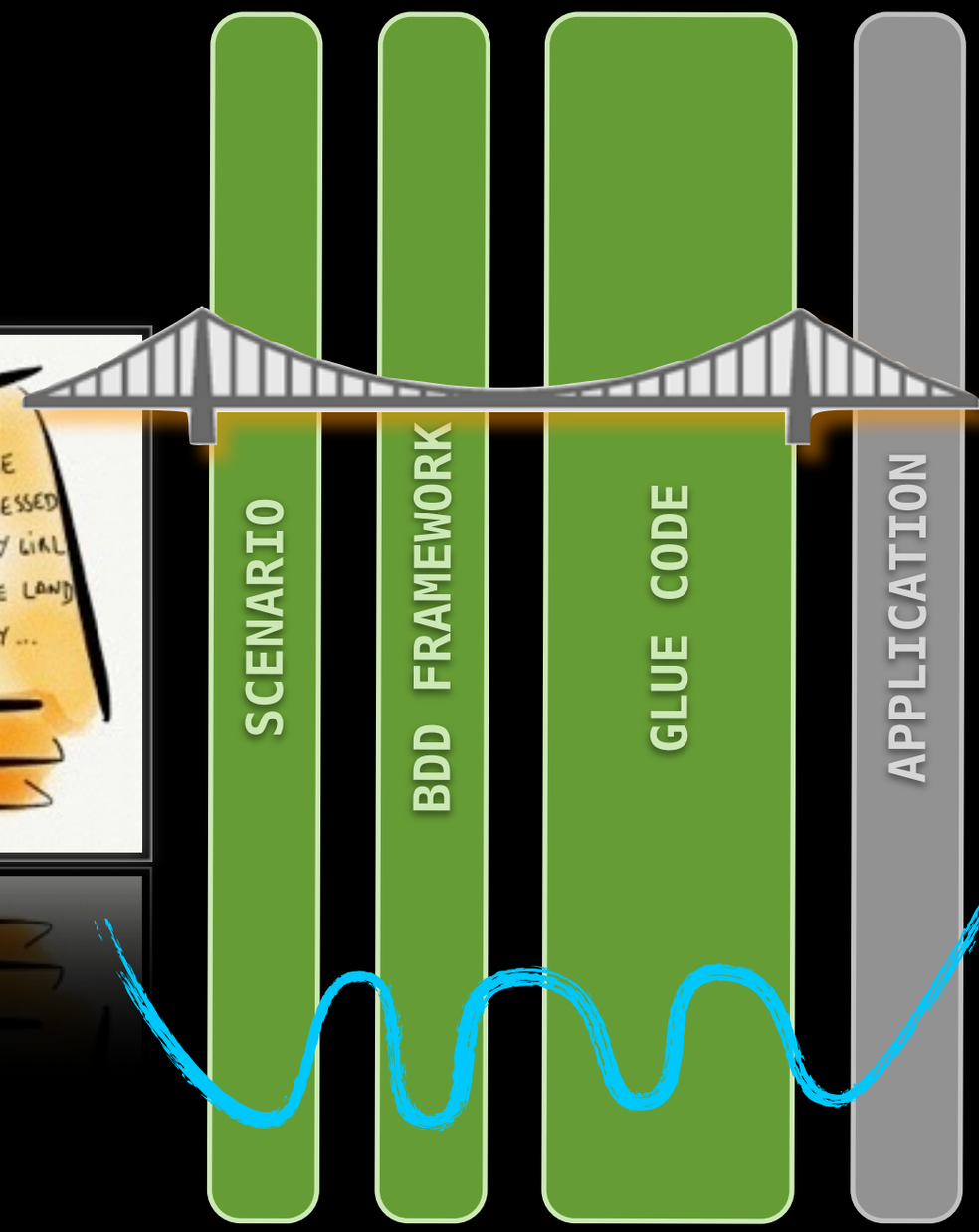
FEST, Selenium, Fluentlenium,
HtmlUnit, Watir...

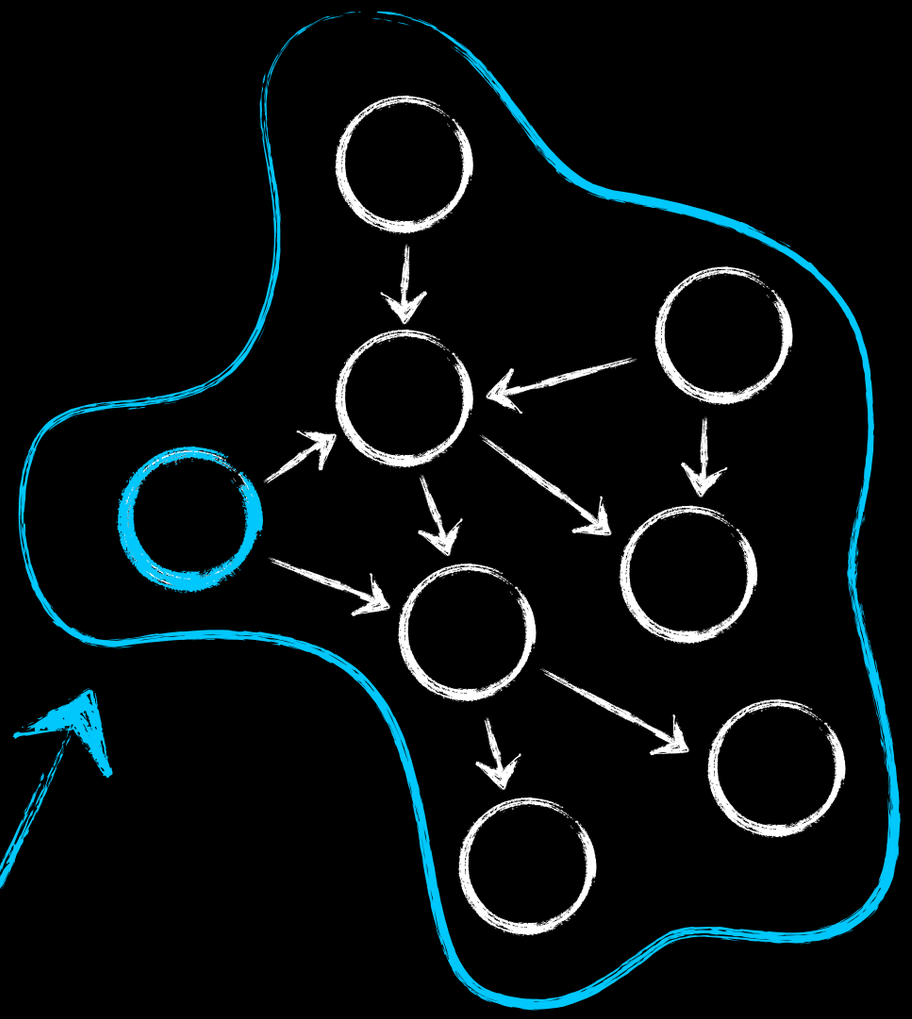




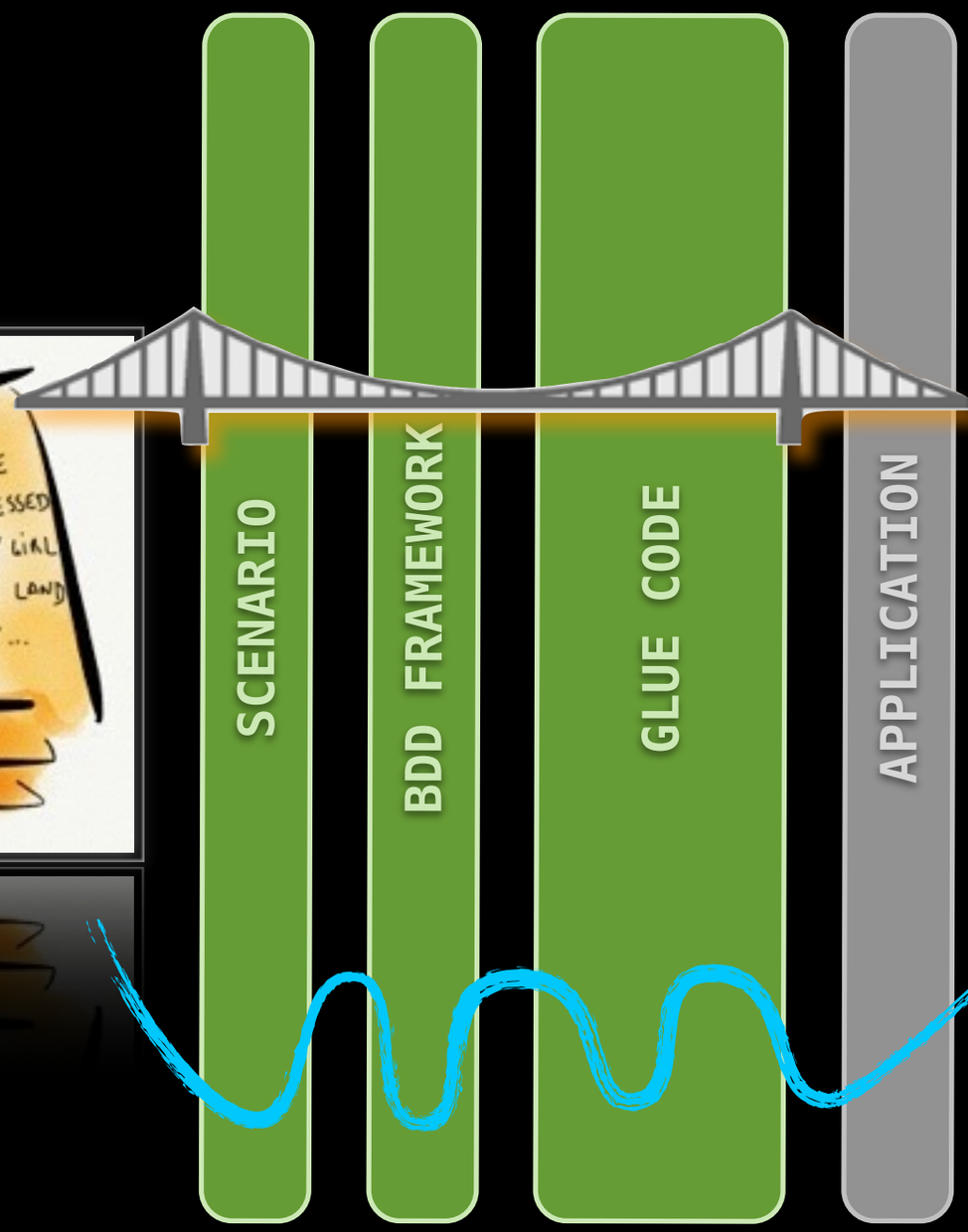
"End to End"

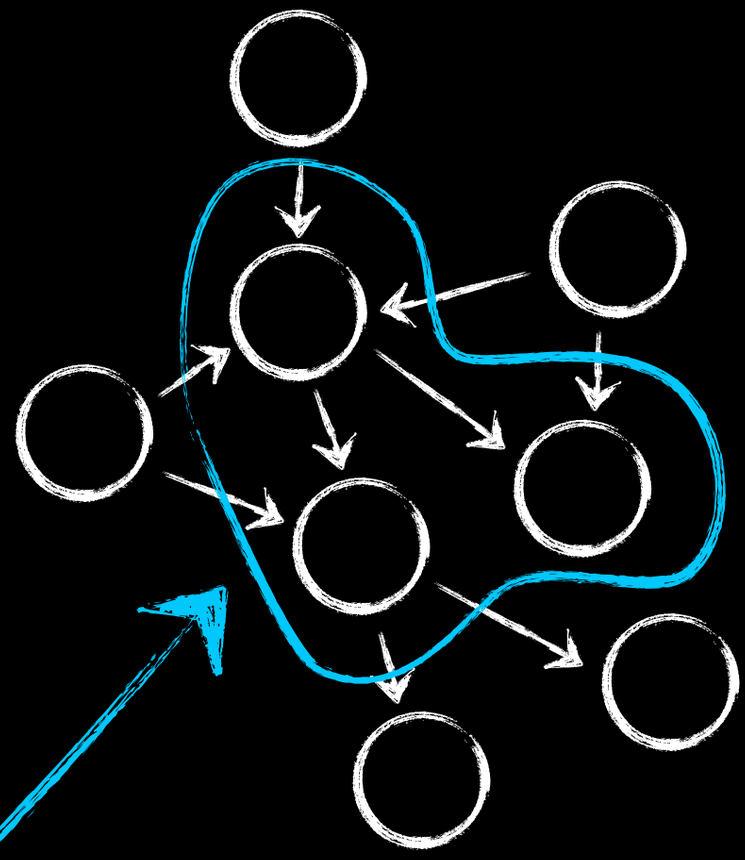
Services
Web, WCF, ...



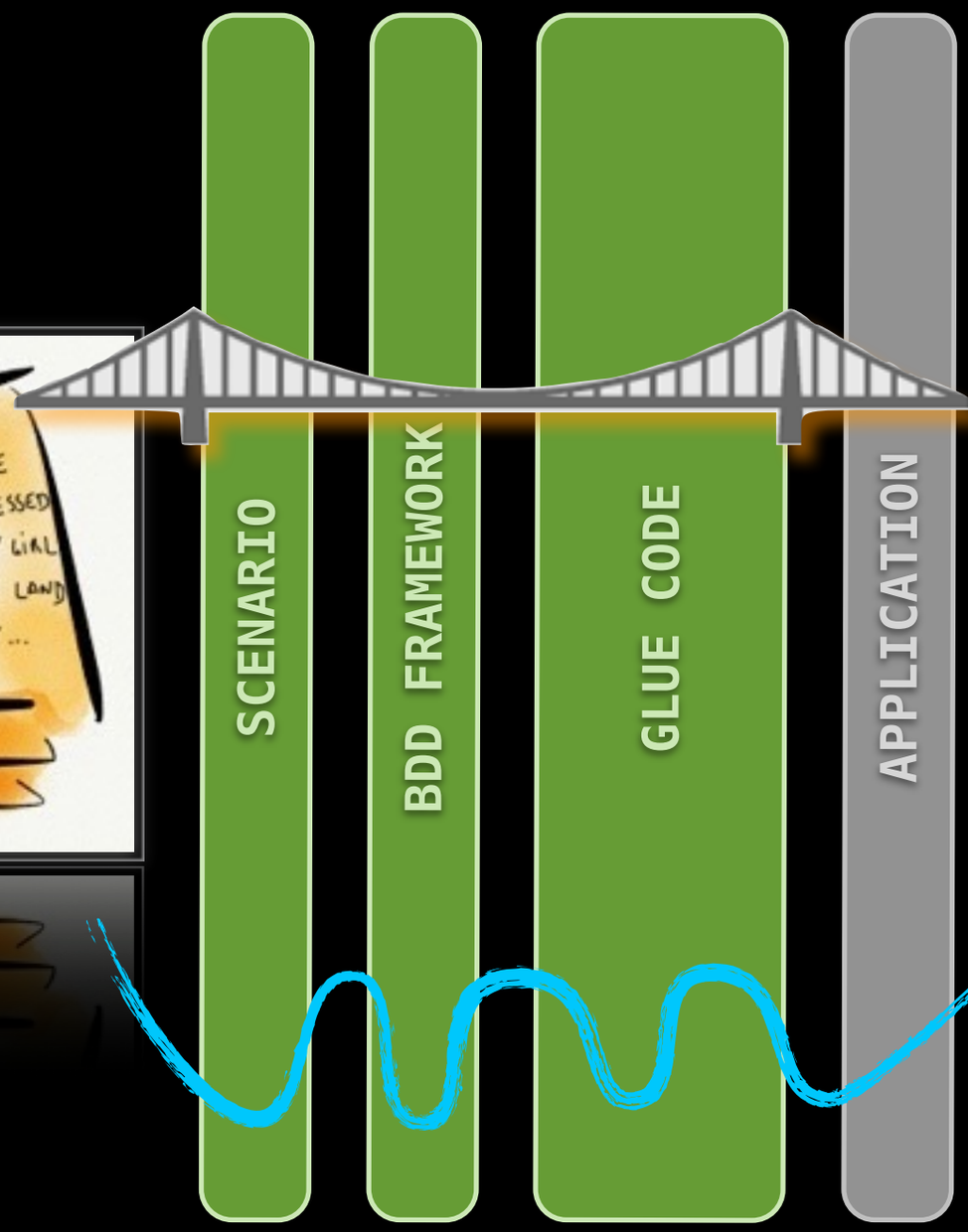


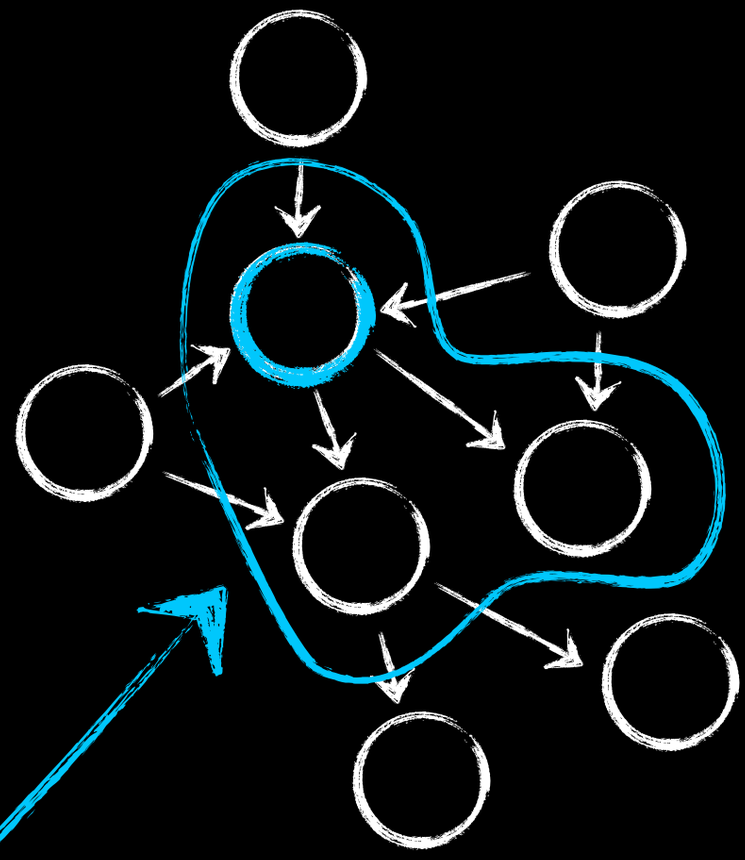
"End to End"



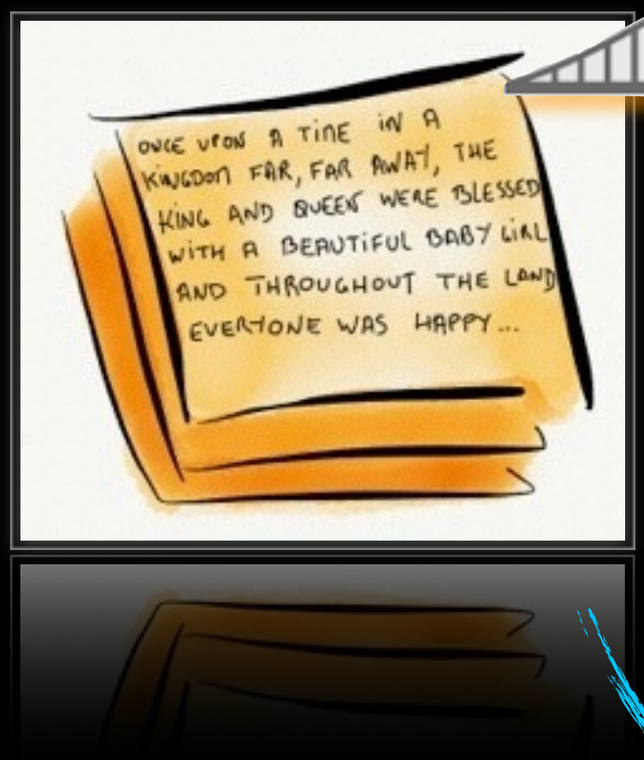


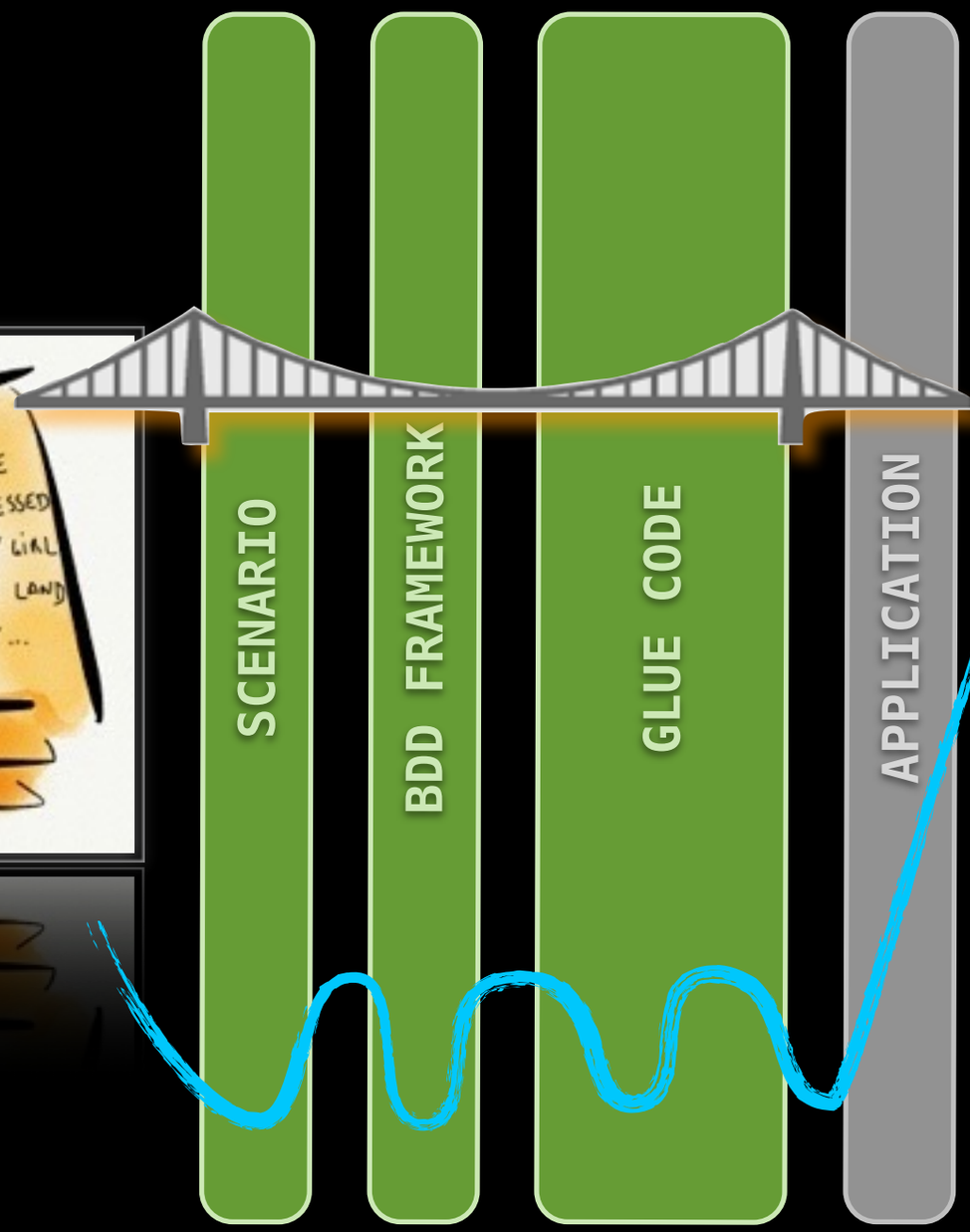
Sub-Module
runtime

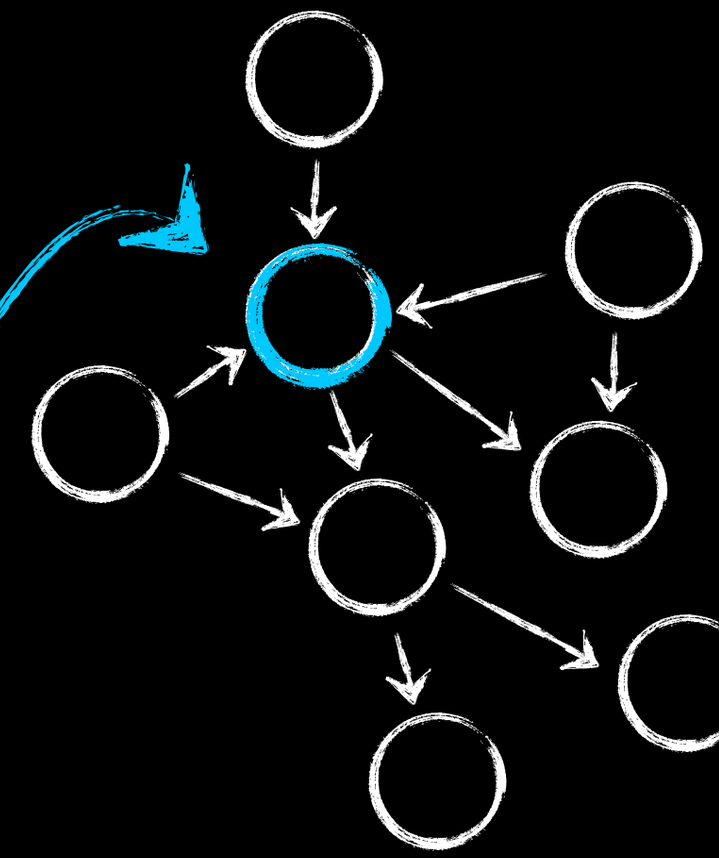




Sub-Module
runtime



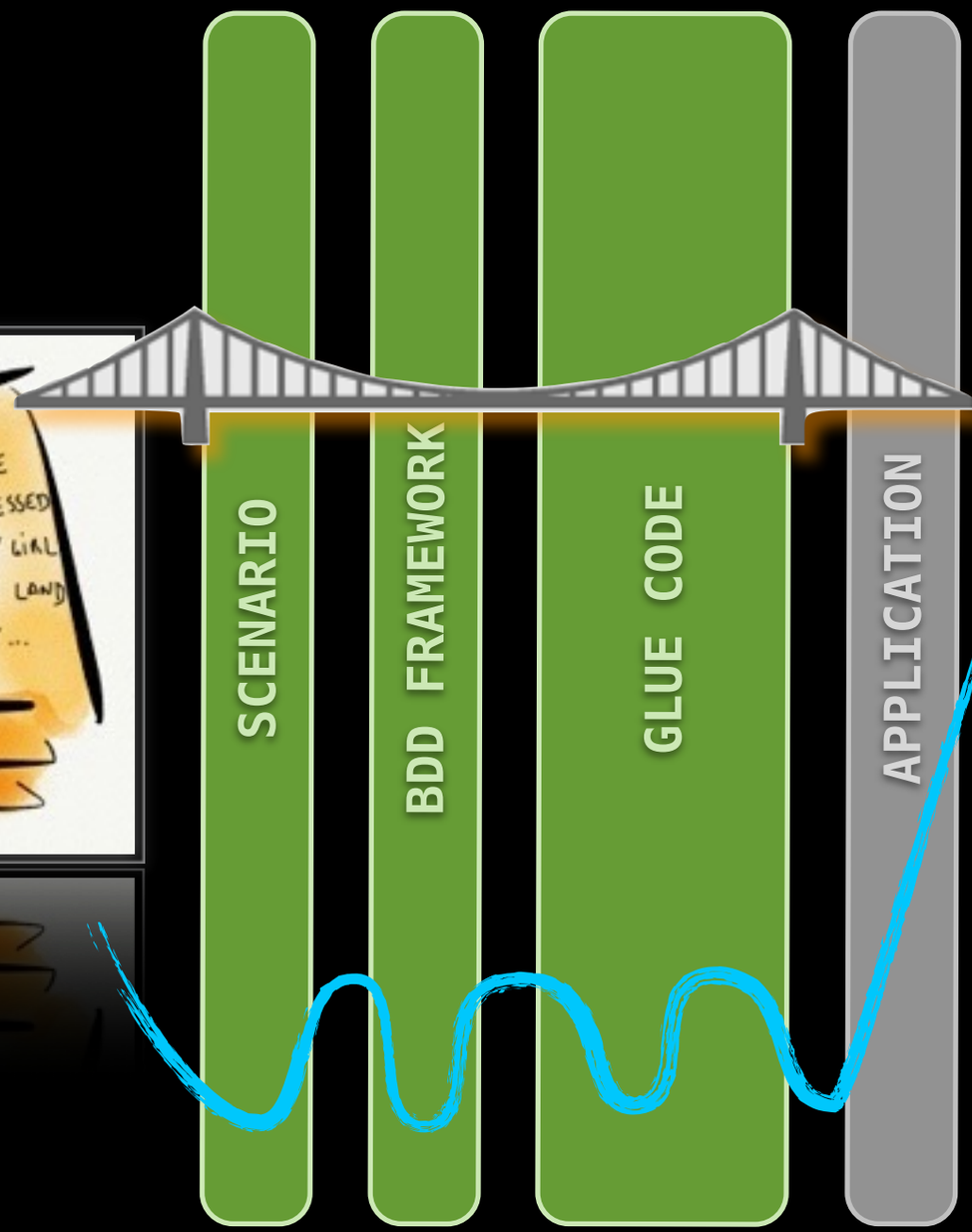


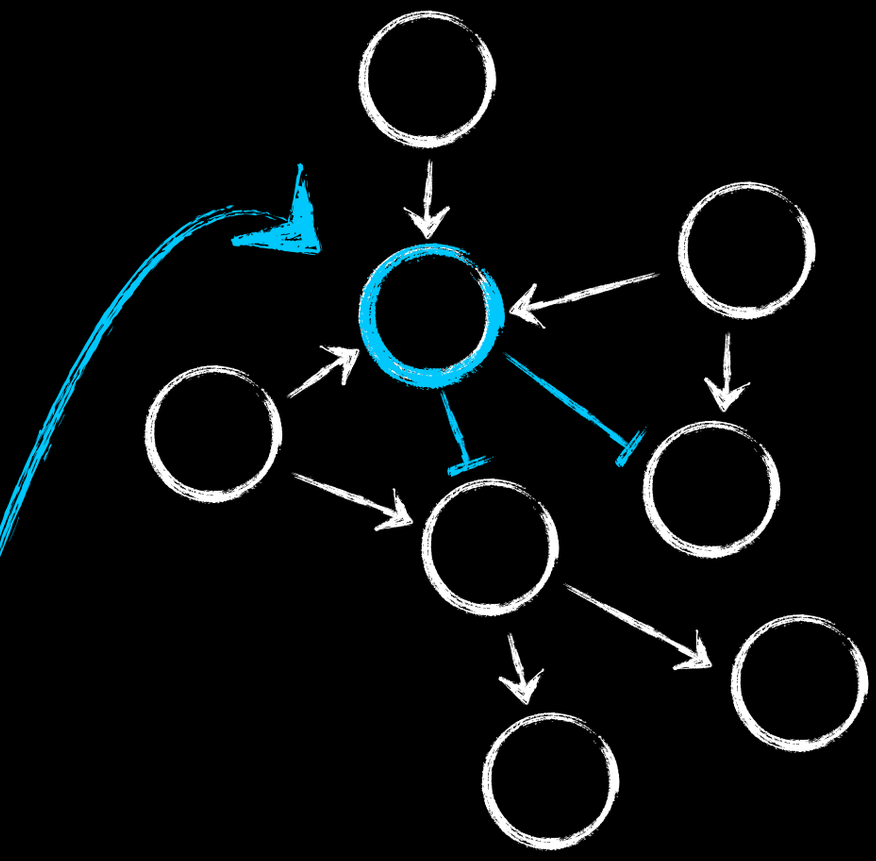


"Quasi-Unitaire"

~~runtime~~

Mock, Stubs...

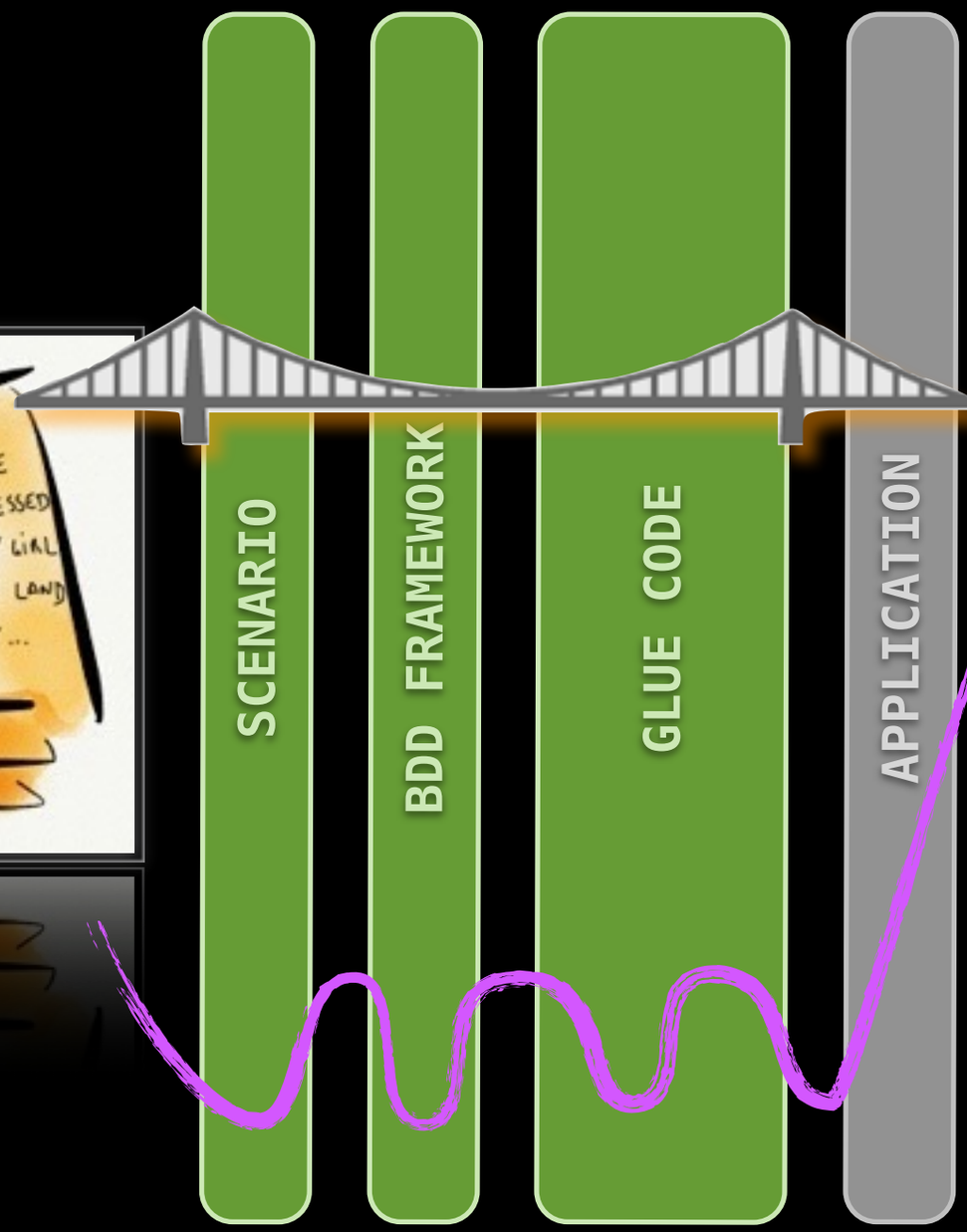


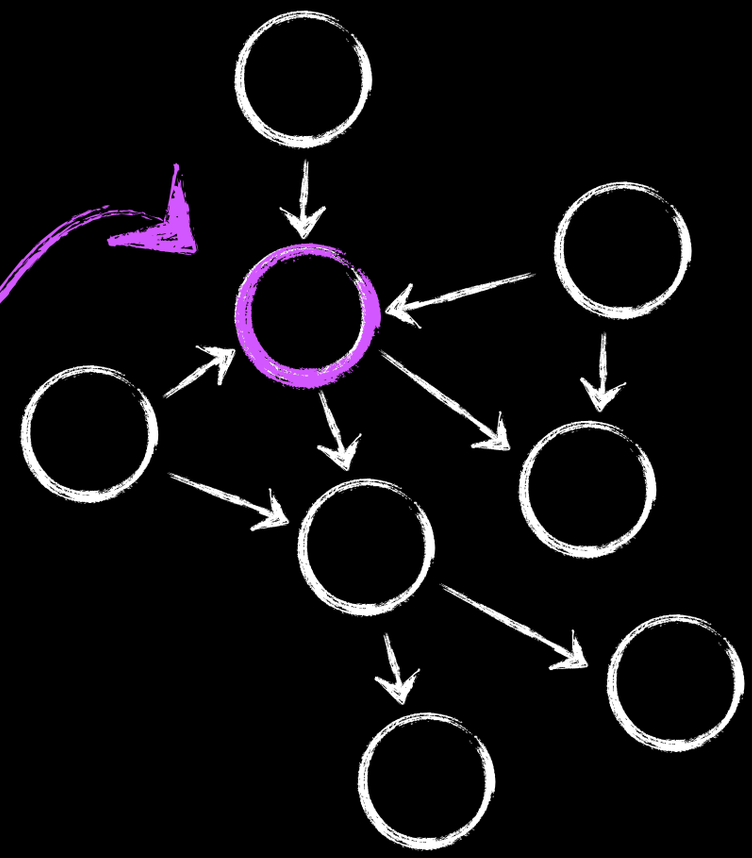


"Quasi-Unitaire"

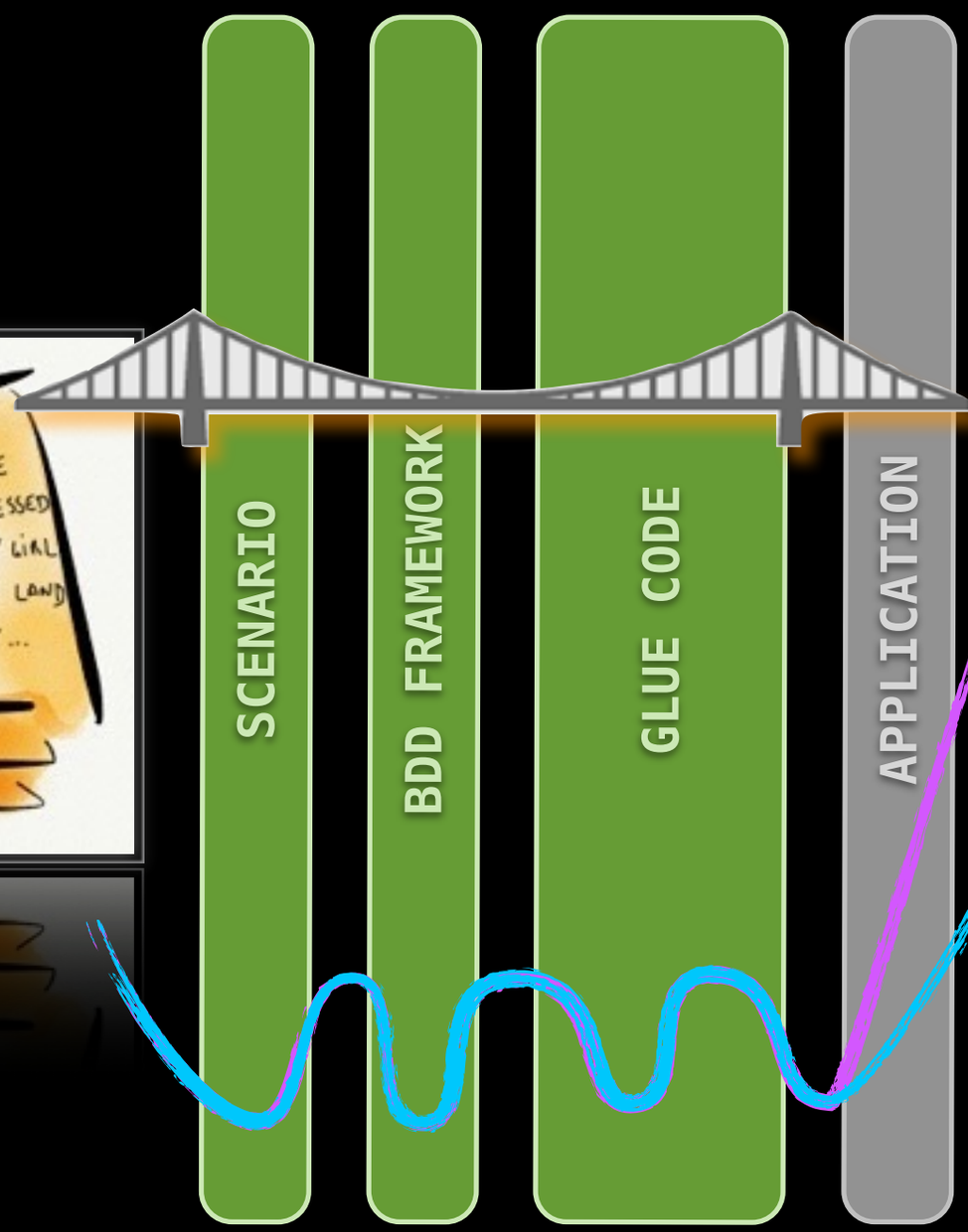
~~runtime~~

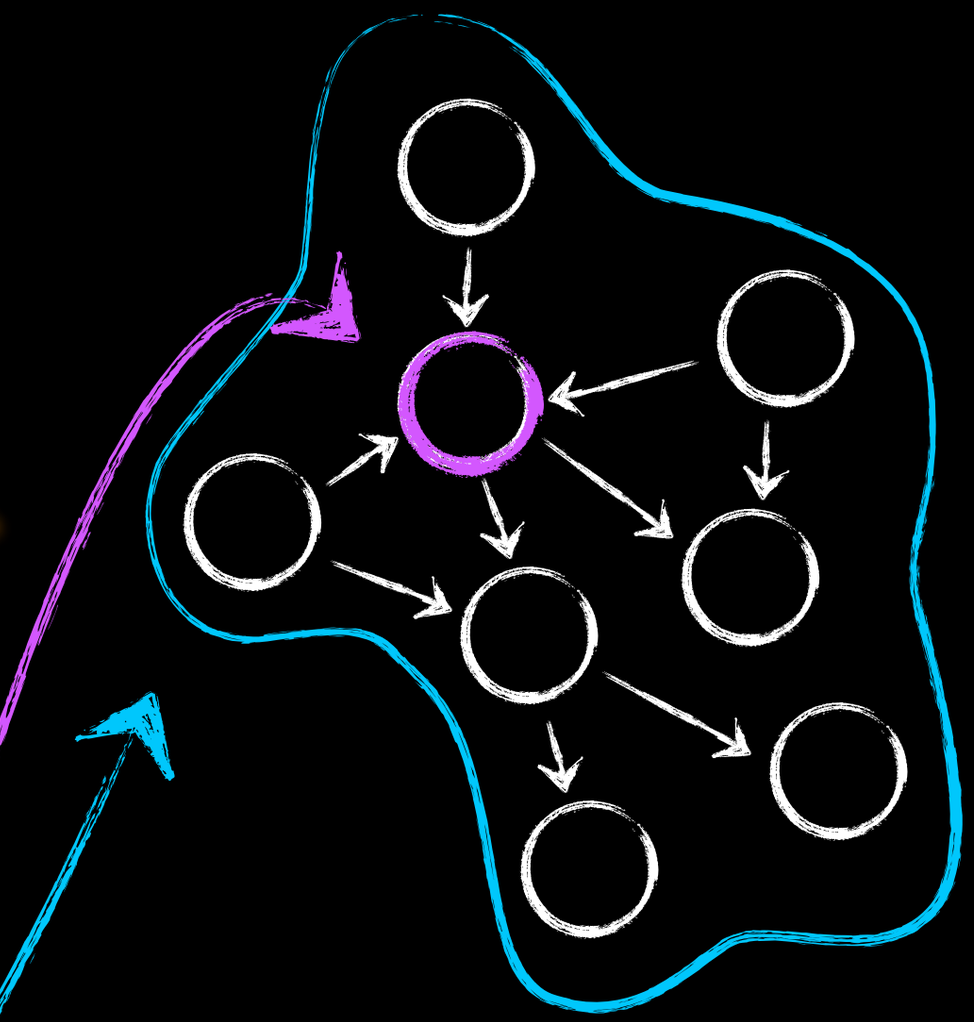

ONCE UPON A TIME IN A
KINGDOM FAR, FAR AWAY, THE
KING AND QUEEN WERE BLESSED
WITH A BEAUTIFUL BABY GIRL
AND THROUGHOUT THE LAND
EVERYONE WAS HAPPY...

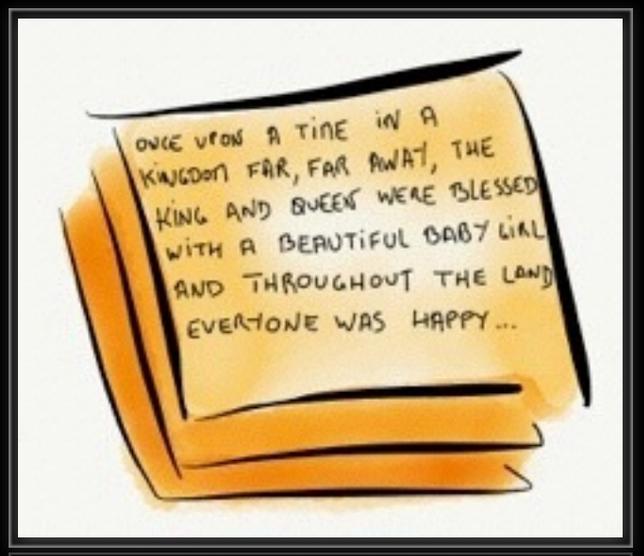

SCENARIO
BDD FRAMEWORK
GLUE CODE
APPLICATION




ONCE UPON A TIME IN A
KINGDOM FAR, FAR AWAY, THE
KING AND QUEEN WERE BLESSED
WITH A BEAUTIFUL BABY GIRL
AND THROUGHOUT THE LAND
EVERYONE WAS HAPPY...

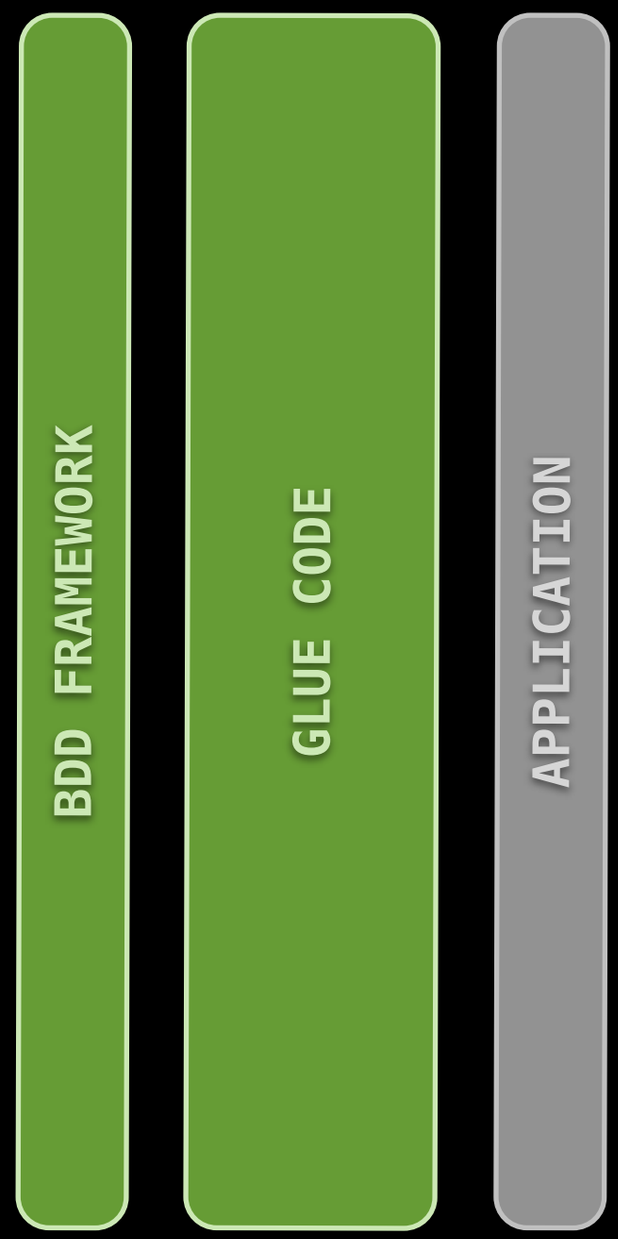

SCENARIO
BDD FRAMEWORK
GLUE CODE
APPLICATION

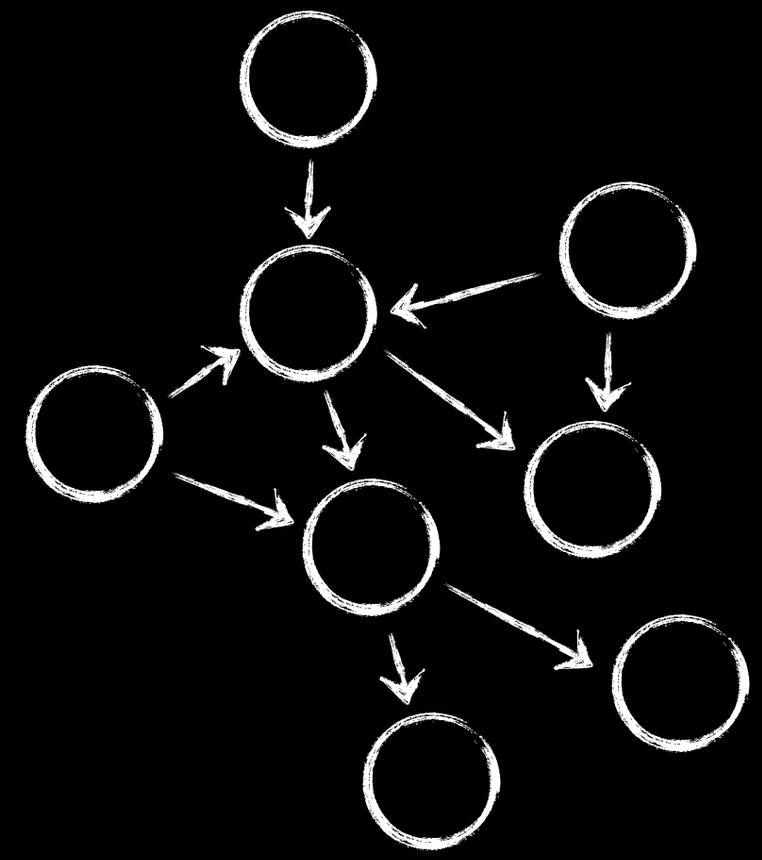




SCENARIO

```
Scenario  Account has sufficient funds
 Given the account balance is 100€
  When the Account Holder requests 20€
  Then the ATM should dispense 20€
   And the account balance should be 80€
   And the card should be returned
```





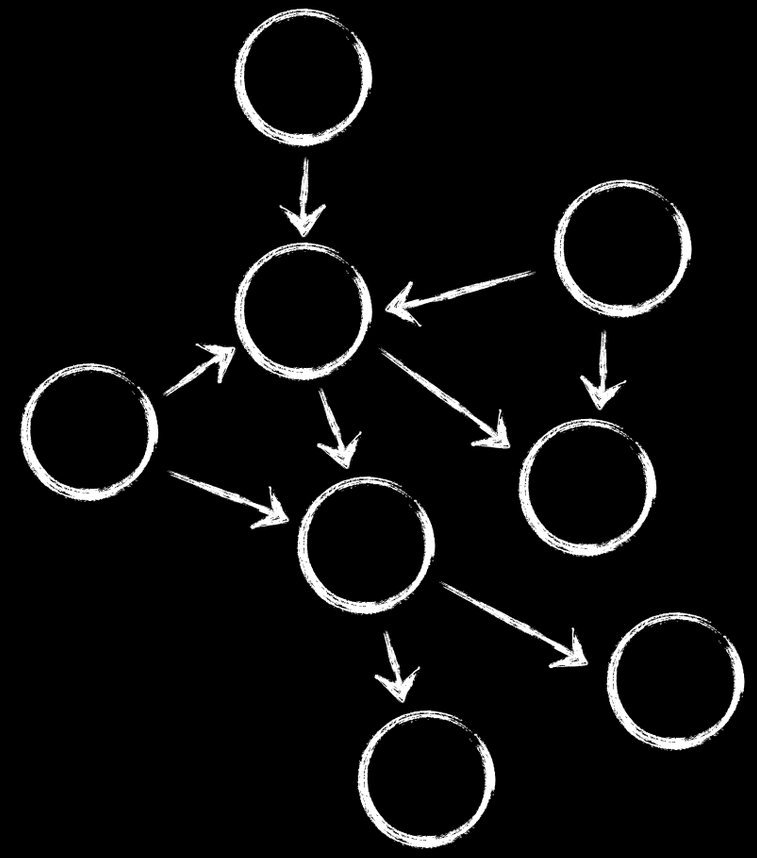
SCENARIO

```
Scenario: Account has sufficient funds
 Given the account balance is 100€
  When the Account Holder requests 20€
  Then the ATM should dispense 20€
   And the account balance should be 80€
   And the card should be returned
```

BDD FRAMEWORK

GLUE CODE

APPLICATION



SCENARIO

```
Scenario: Account has sufficient funds
 Given the account balance is 100€
  When the Account Holder requests 20€
  Then the ATM should dispense 20€
   And the account balance should be 80€
   And the card should be returned
```

BDD FRAMEWORK

GLUE CODE

```
@Given("^the account balance is (\d+)€$")
public void defineAccountBalanceInEuro(BigDecimal
balance) {
  throw new PendingException("Implements me!");
}

@When("^the Account Holder request (\d+)€$")
public void withdrawInEuro (BigDecimal amount) {
  throw new PendingException("Implements me!");
}

@Then("^the ATM should dispense (\d+)€$
public void assertMoneyDispensedInEuro (BigDecimal
amount) {
  throw new PendingException("Implements me!");
}


@Then("^the account balance should be (\d+)€$")
public void assertBalanceInEuro(BigDecimal amount) {
  throw new PendingException("Implements me!");
}
```

APPLICATION

SCENARIO

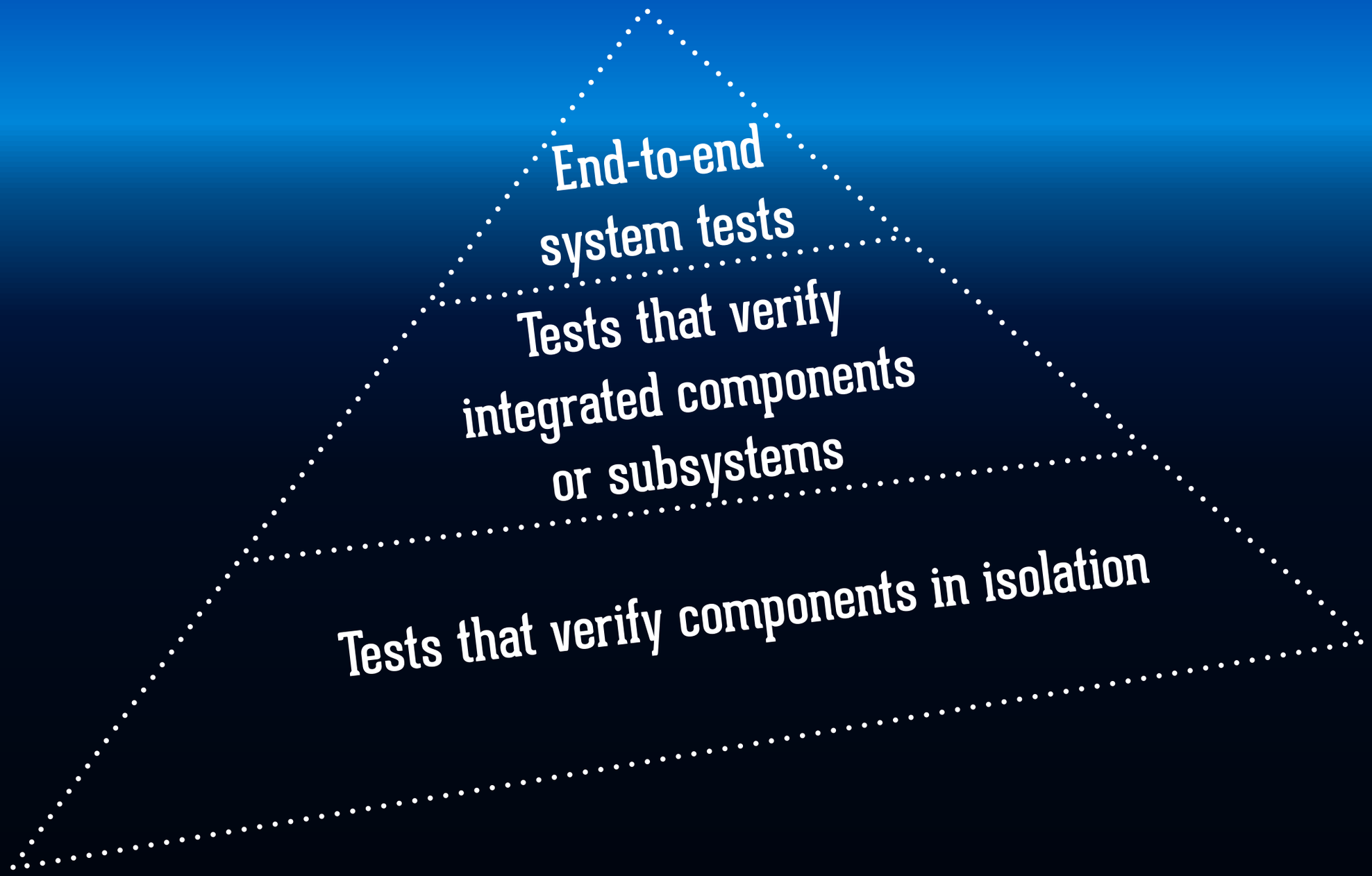
```
Scenario: Account has sufficient funds
 Given the account balance is 100€
  When the Account Holder requests 20€
  Then the ATM should dispense 20€
   And the account balance should be 80€
   And the card should be returned
```

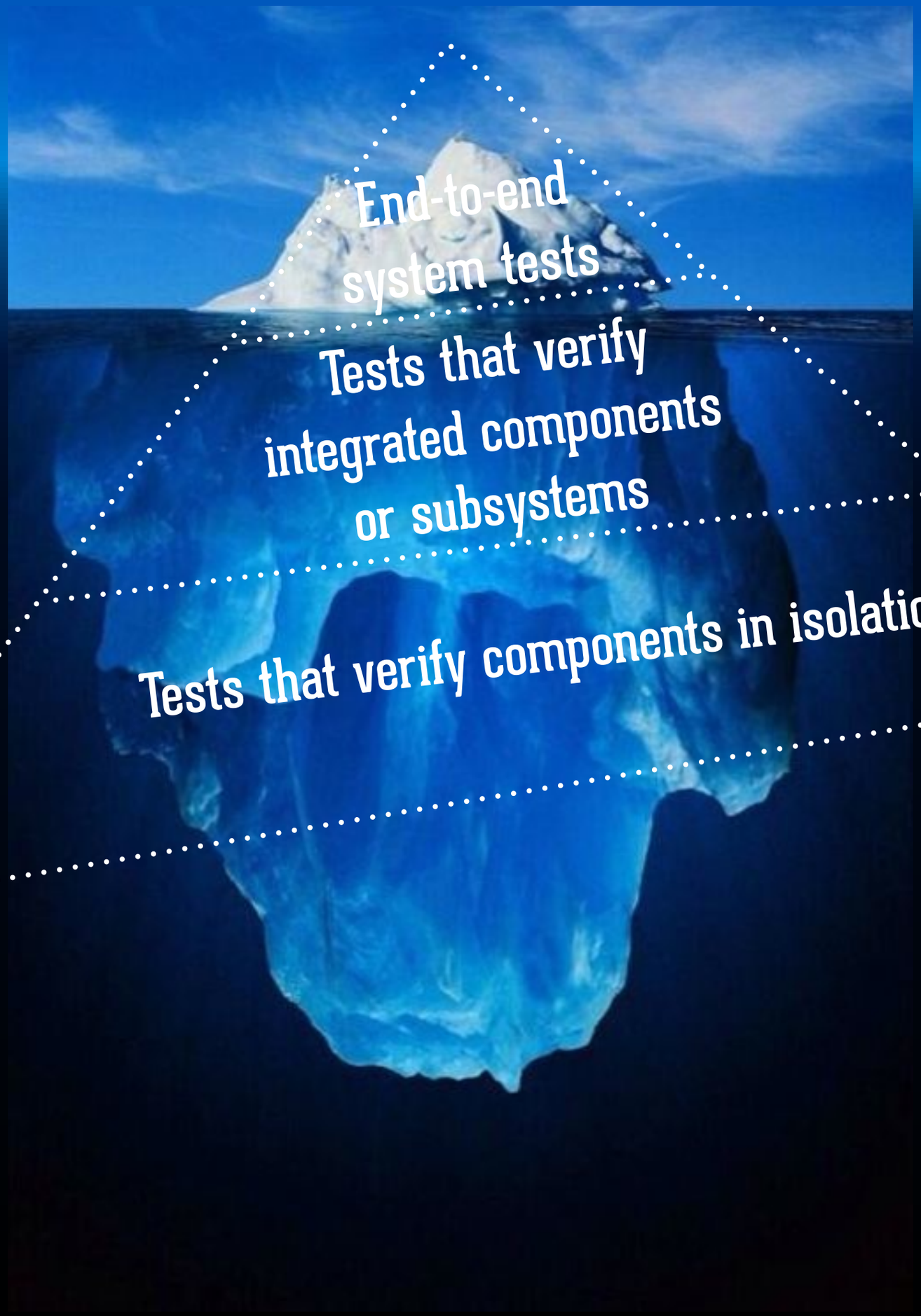
BDD FRAMEWORK

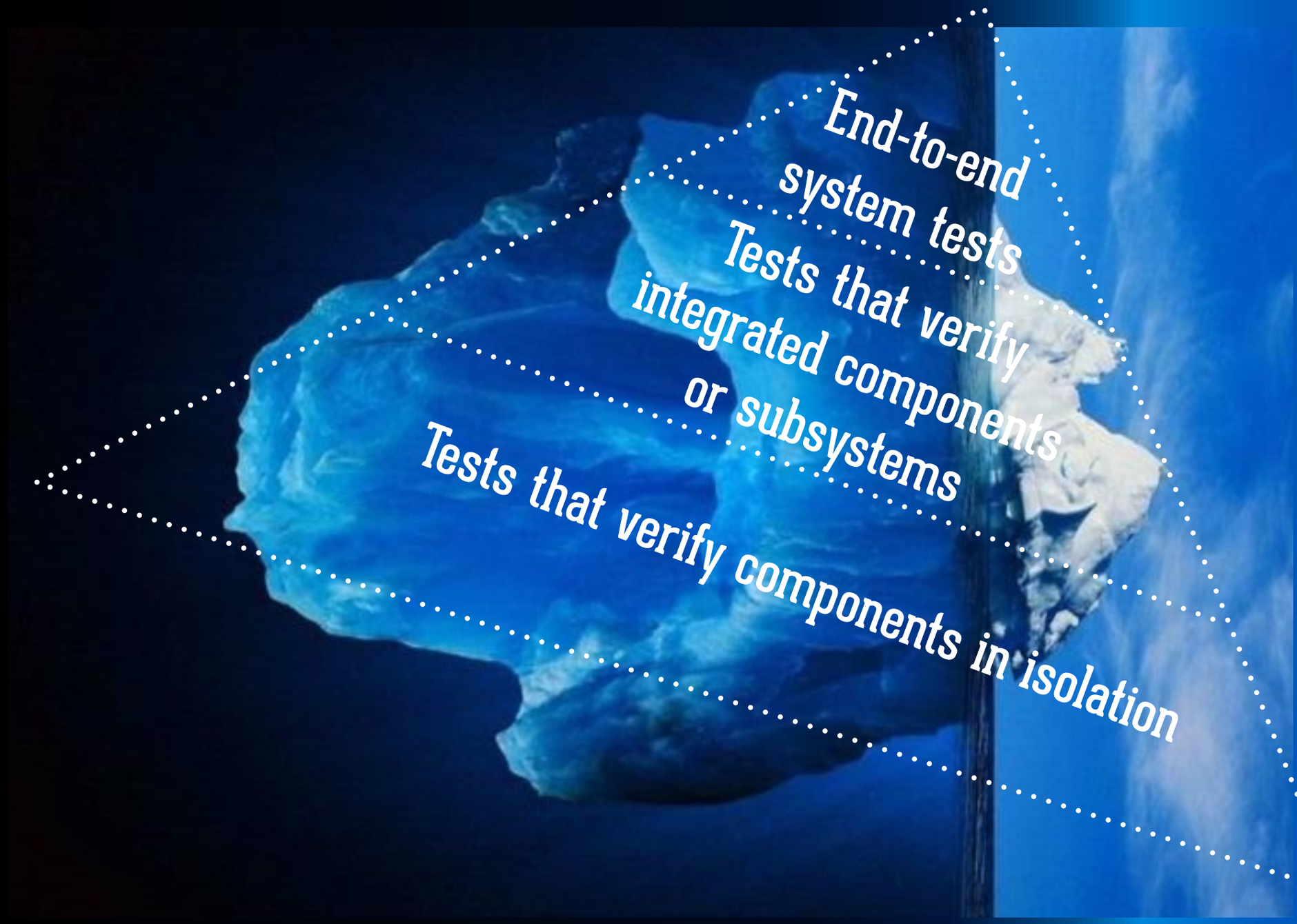
GLUE CODE

```
@Given("^the account balance is (\d+)€$")
public void defineAccountBalanceInEuro(BigDecimal
balance) {
  account().setBalance(euro(balance));
}

@When("^the Account Holder request (\d+)€$")
public void withdrawInEuro (BigDecimal amount) {
  atm().withdraw(account(), euro(amount));
}

@Then("^the ATM should dispense (\d+)€$
public void assertMoneyDispensedInEuro (BigDecimal
amount) {
  TransactionLog txLog = atm().transactionLog();
  Money dispensed = txLog.lastAmountDispensed();
  assertThat(dispensed).isEqualTo(euro(amount));
}

@Then("^the account balance should be (\d+)€$")
public void assertBalanceInEuro(BigDecimal amount) {
  Money actualBalance = account().balance();
  assertThat(actualBalance).isEqualTo(euro(amount));
}
```

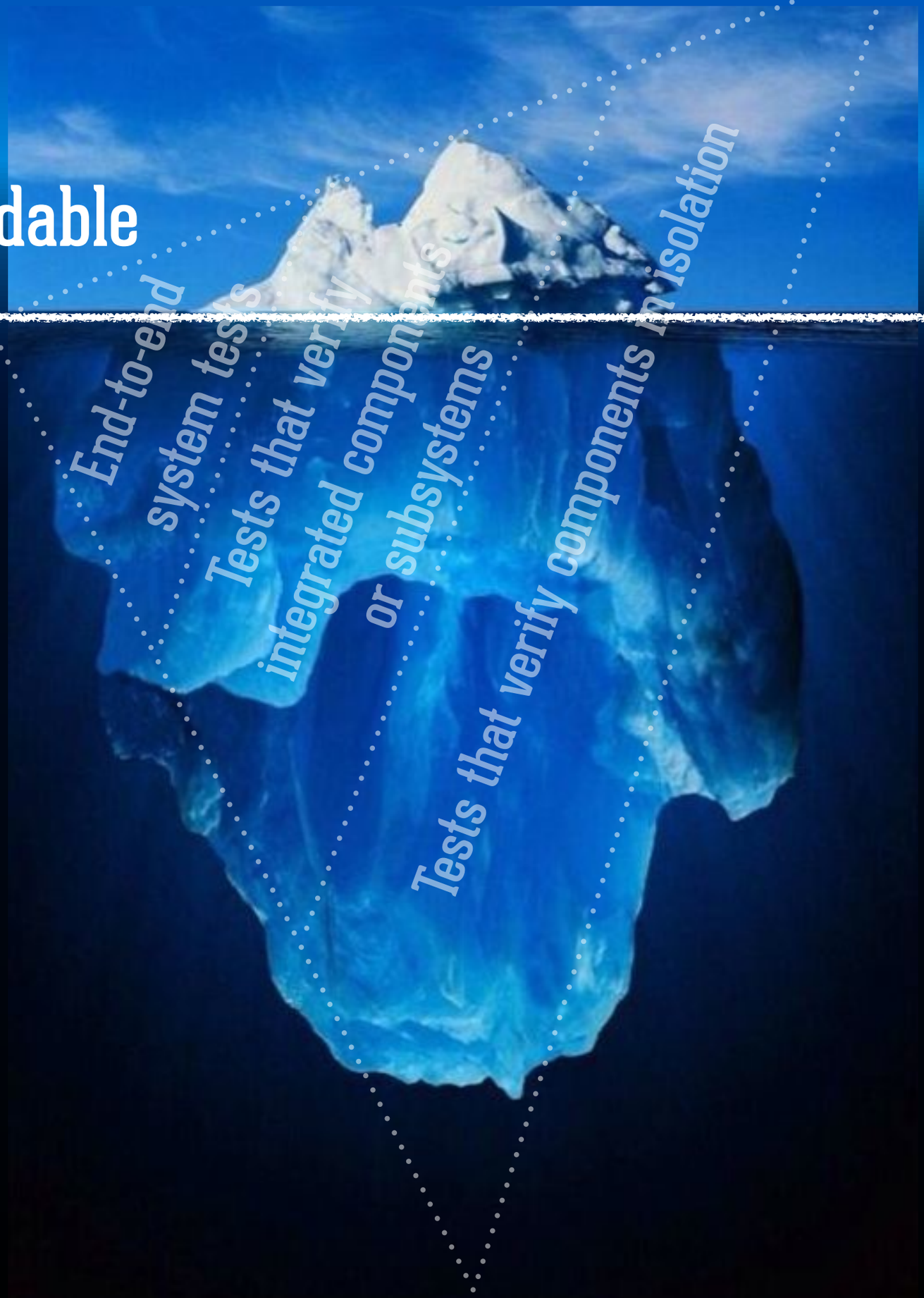
APPLICATION


End-to-end
system tests
Tests that verify
integrated components
or subsystems
Tests that verify components in isolation


End-to-end
system tests
Tests that verify
integrated components
or subsystems
Tests that verify components in isolation

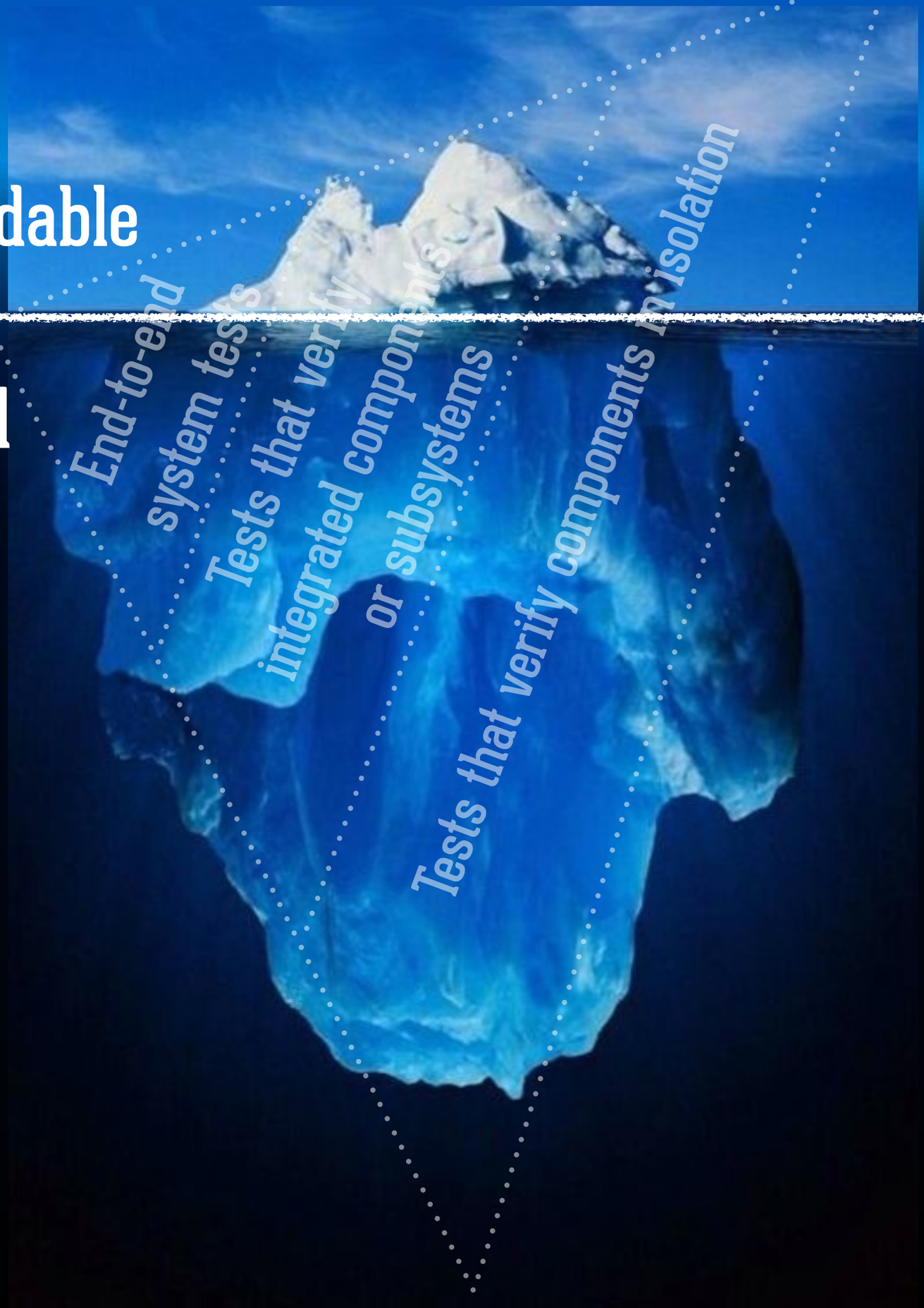

End-to-end
system tests
Tests that verify
integrated components
or subsystems
Tests that verify components in isolation

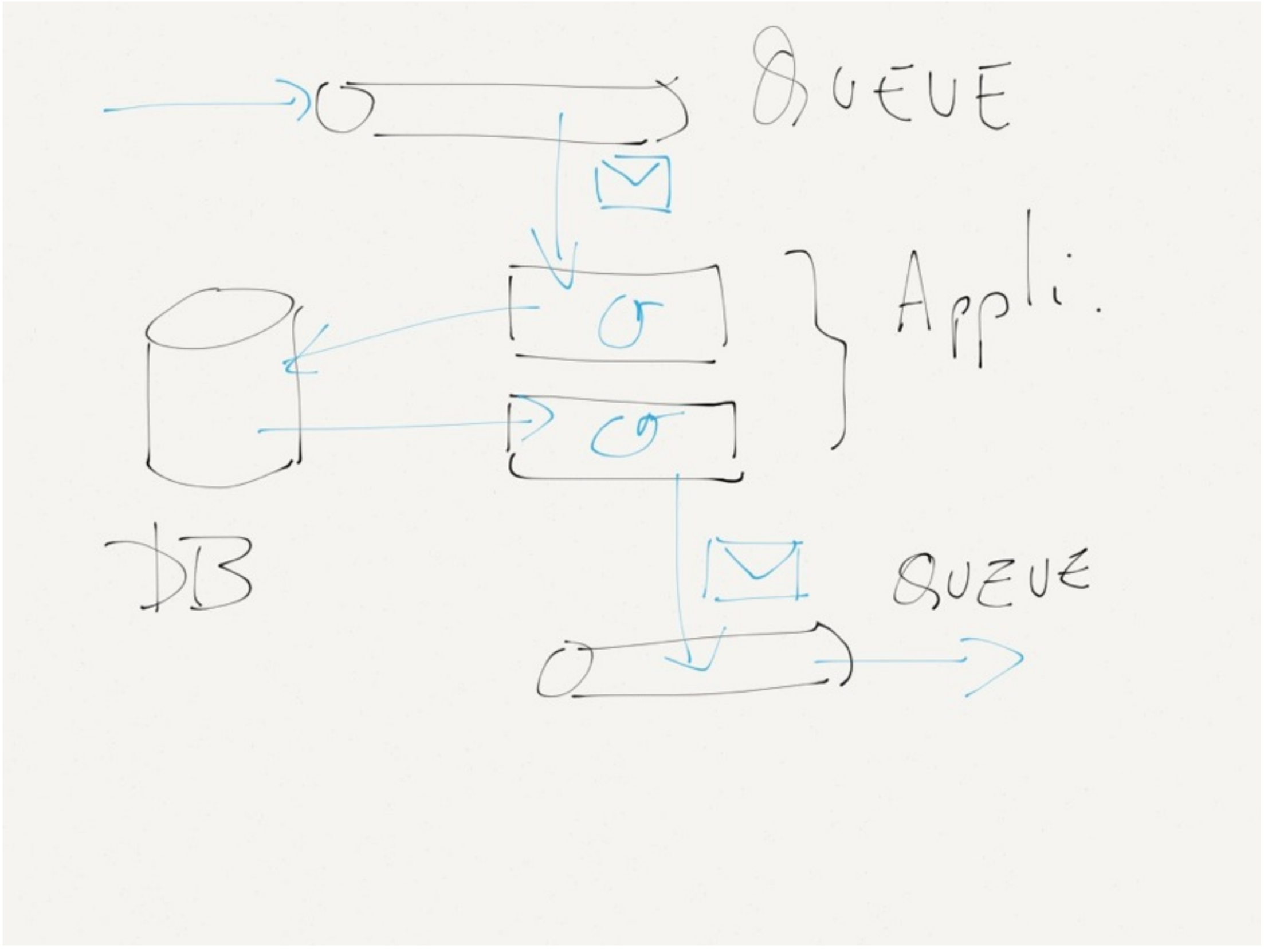
Business Readable

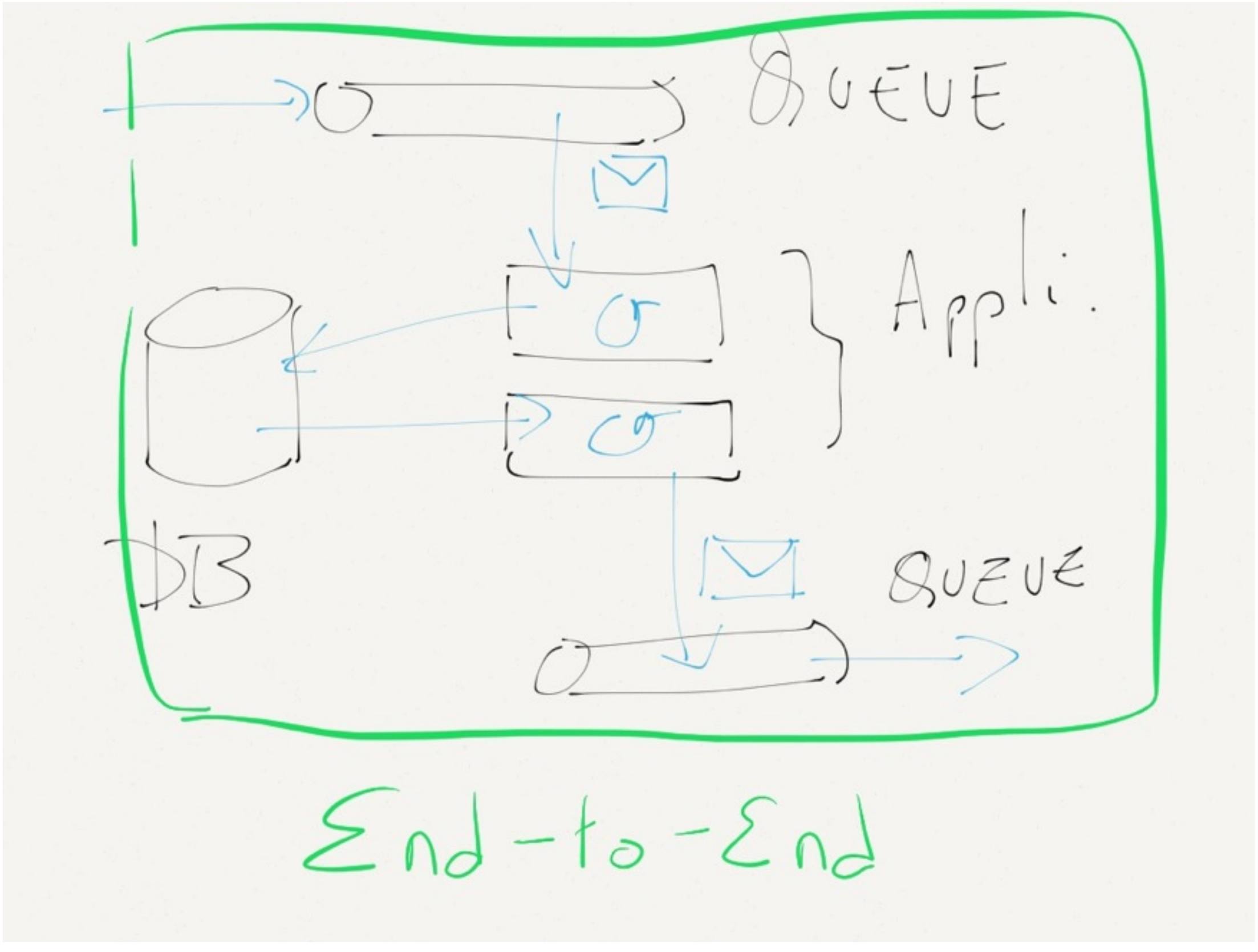


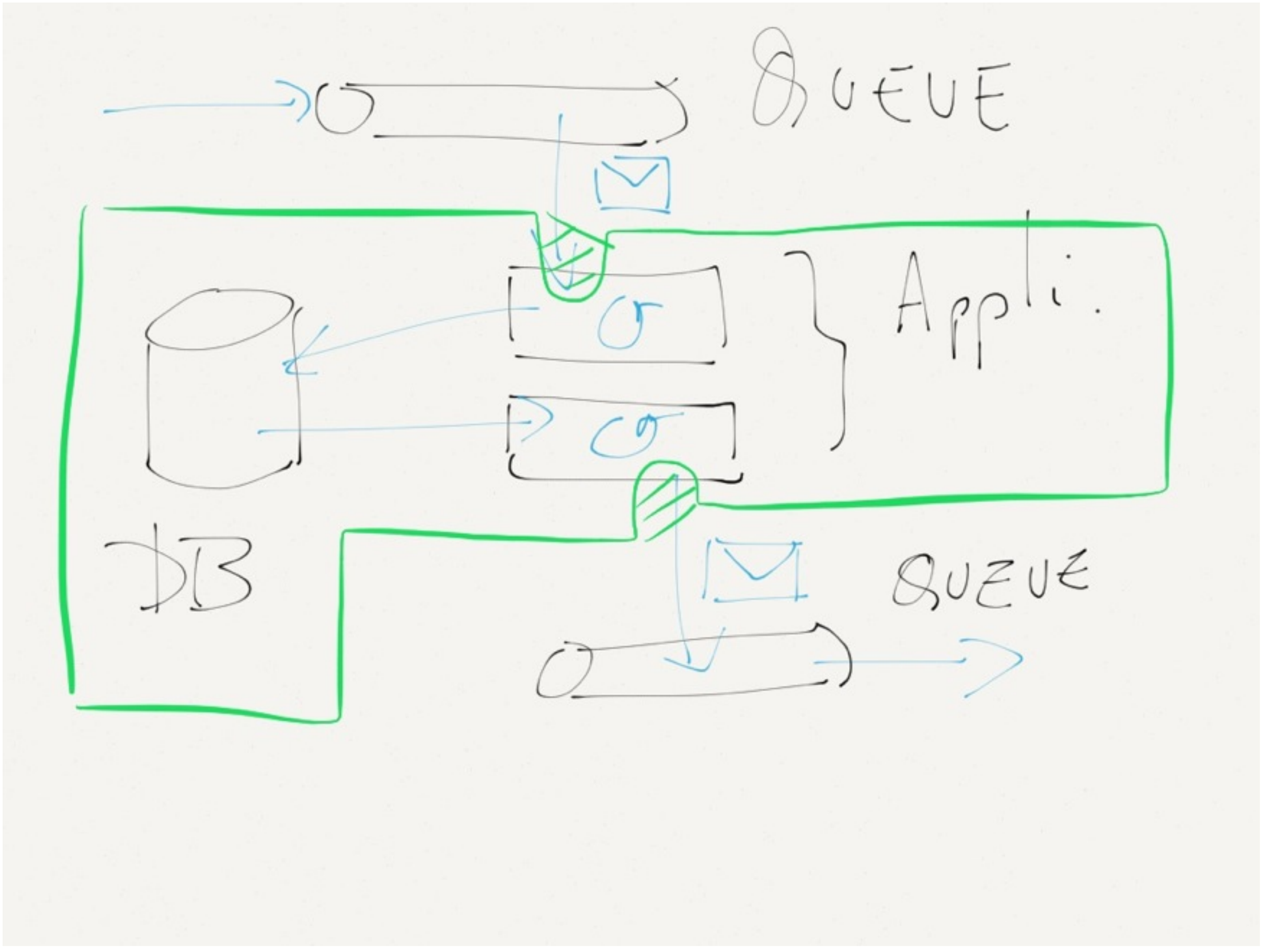
Business Readable

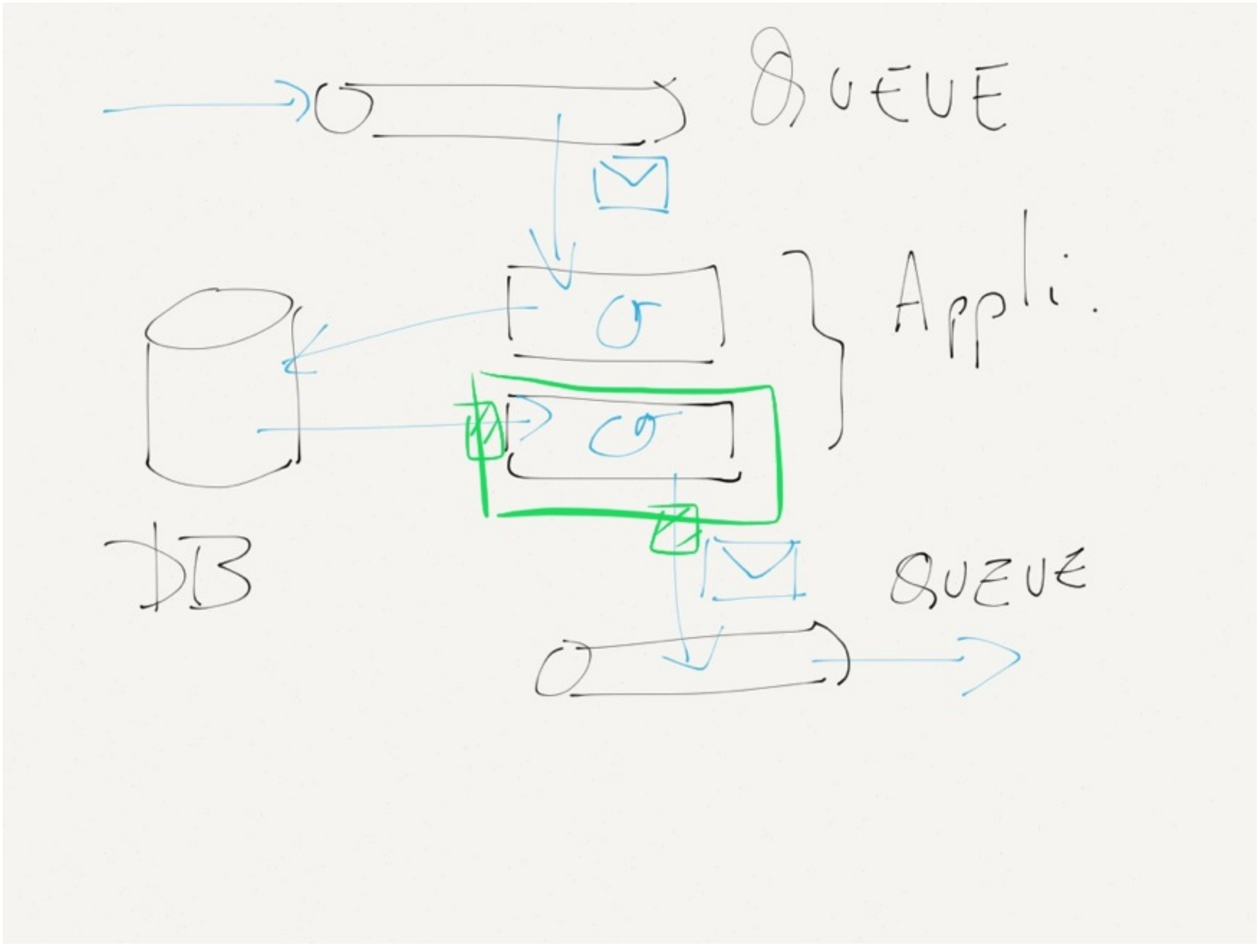
Technical

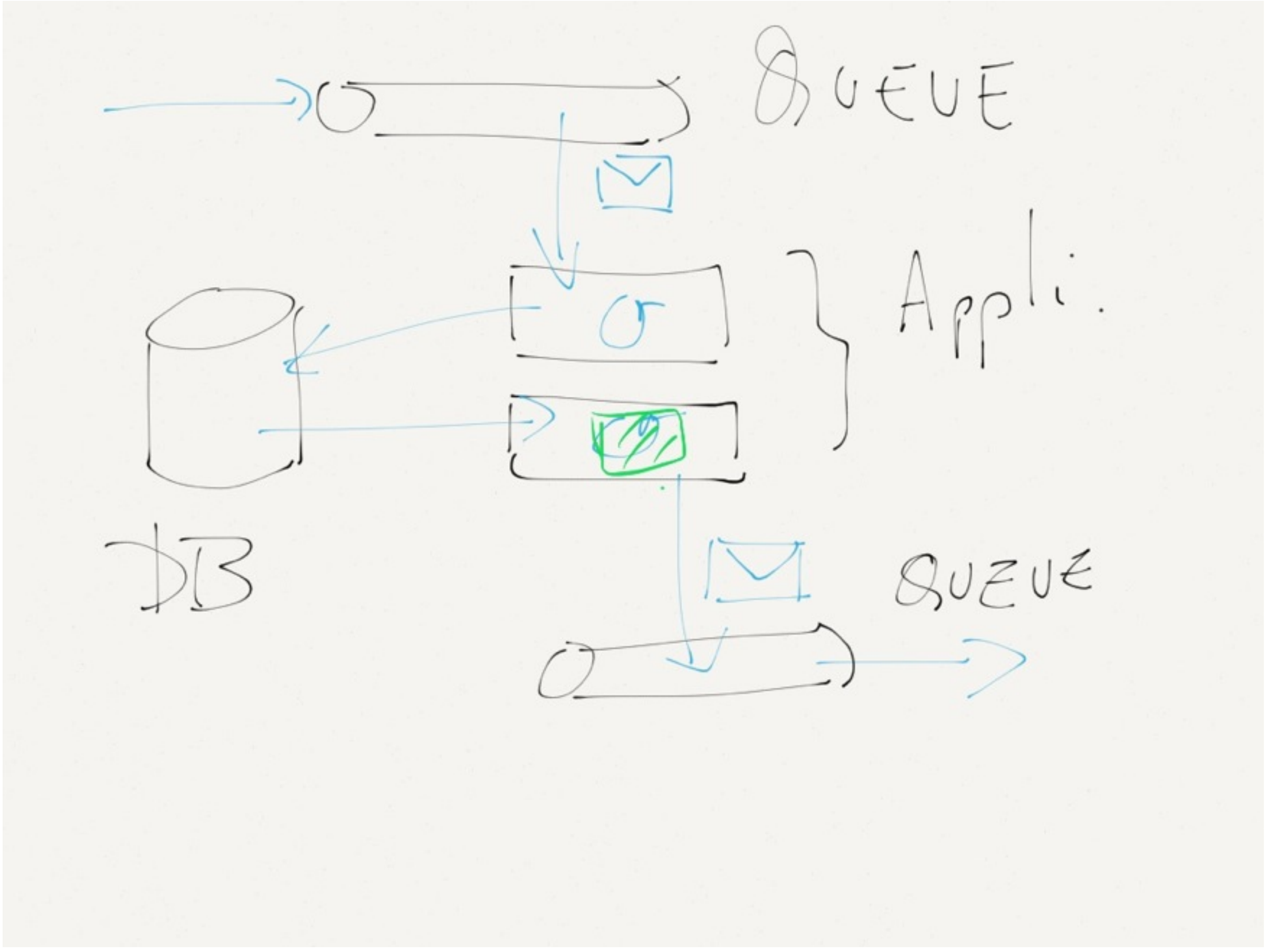



QUEUE
Appli.
DB
QUEUE


QUEUE
Appli.
DB
QUEUE
End-to-End

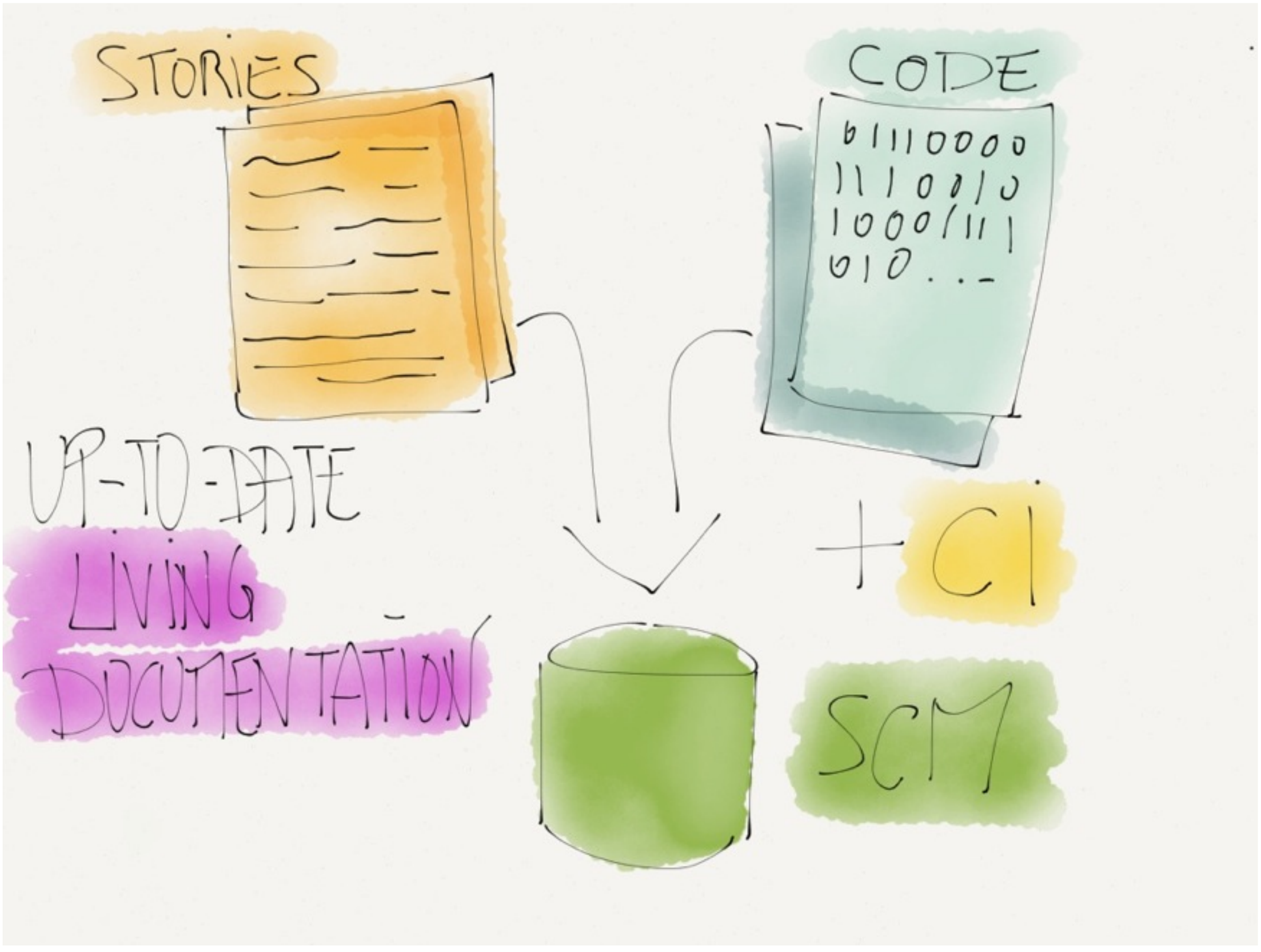

QUEUE
Appli.
DB
QUEUE


QUEUE
Appli.
DB
QUEUE

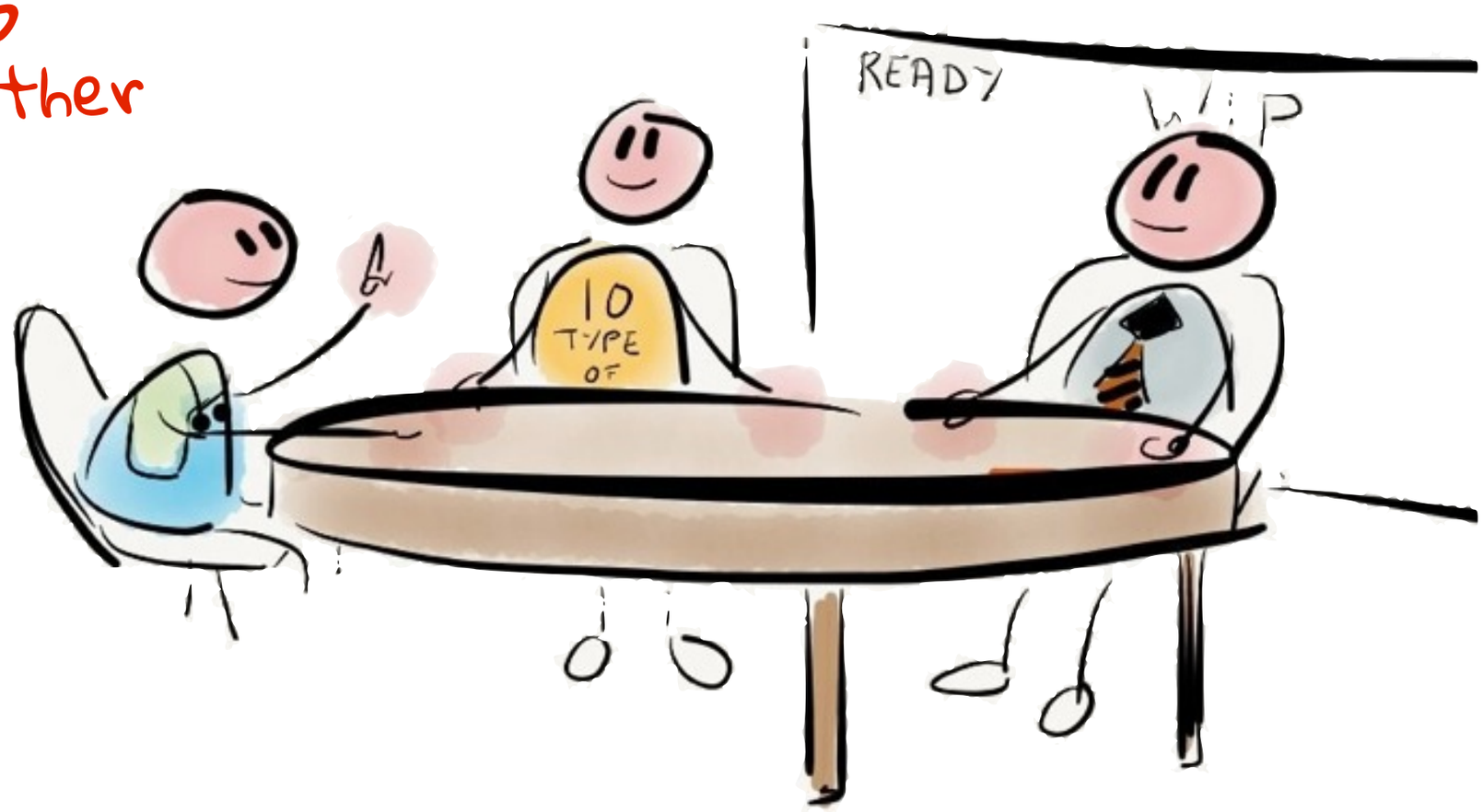

QUEUE
Appli.
DB
QUEUE

STEPS

STEP BY STEP


STORIES
CODE
0b11100000
11101010
10001111
010...
UP-TO-DATE
LIVING
DOCUMENTATION
+CI
SCM

Discuss Together



Discuss Together

Share

Discuss Together

Share

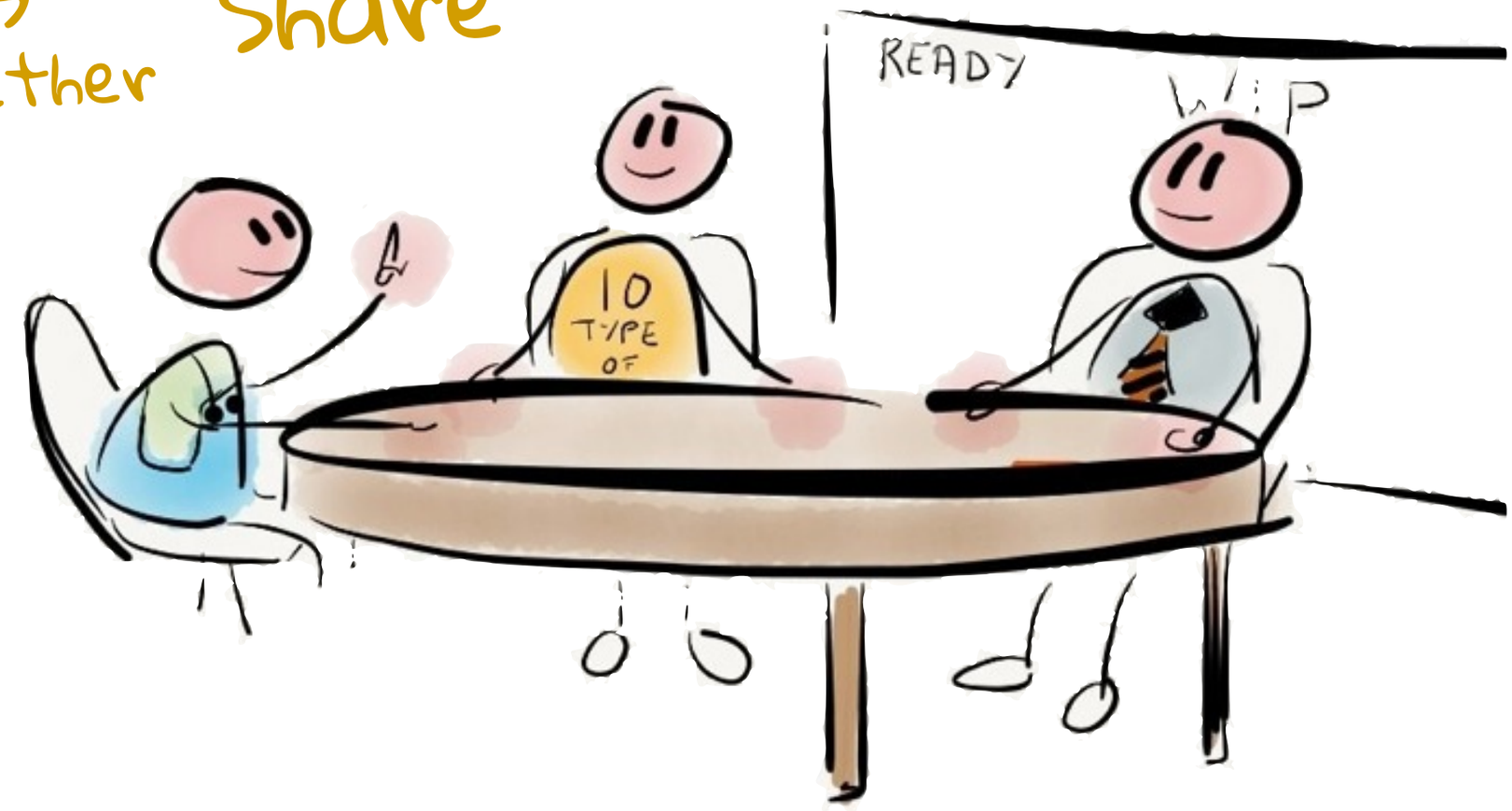
Clarify

Examples

Scenario

Discuss Together

Share



Clarify

Examples

Scenario

Elaborate

Requirements

Discuss Together

Share

Clarify

Examples

Scenario

Elaborate

Requirements

Verify

Can become

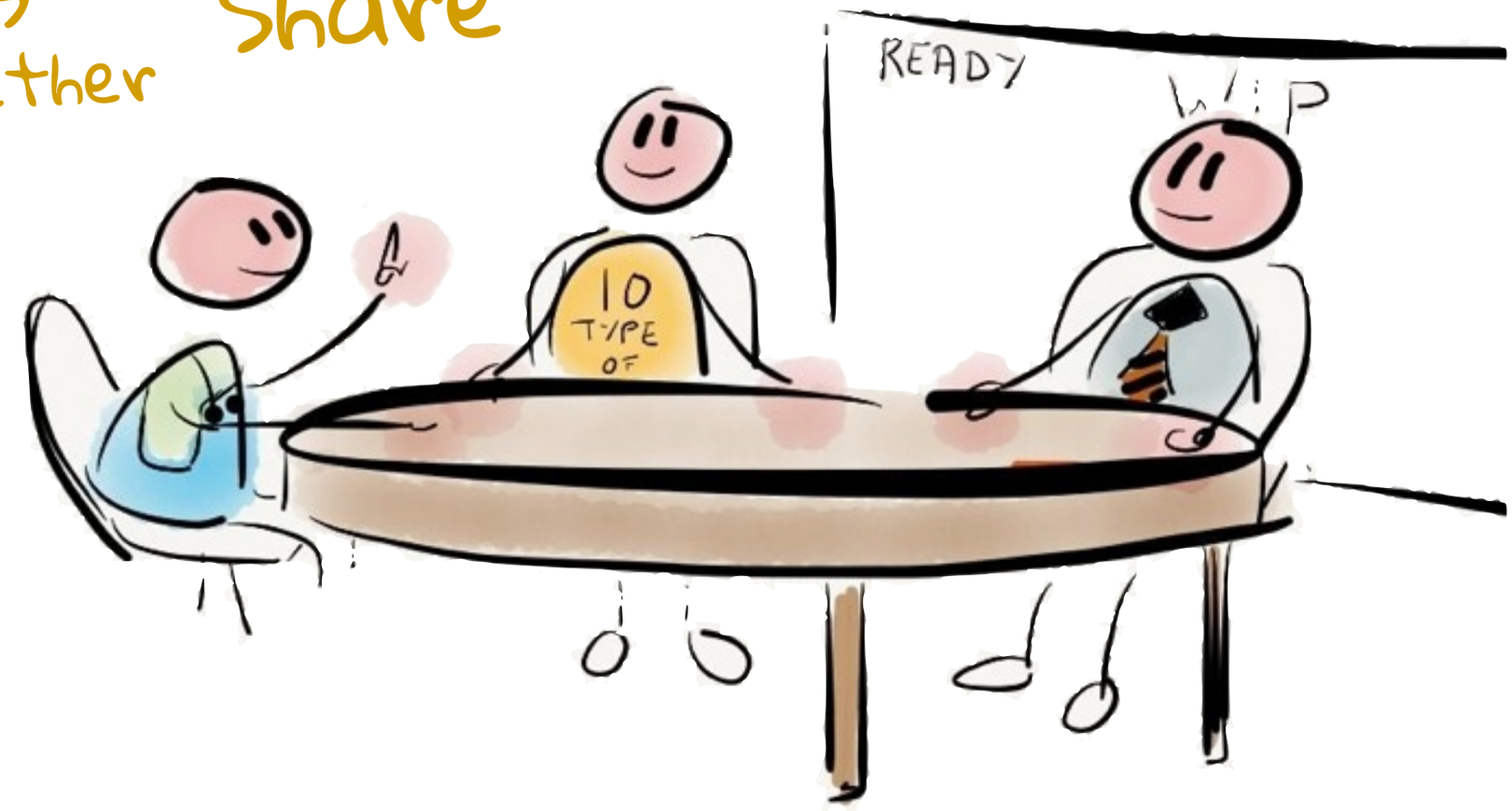
Executable

Automation

Discuss

Together

Share



Clarify

Examples

Scenario

Elaborate

Requirements

Verify

Can become

Executable

Automation

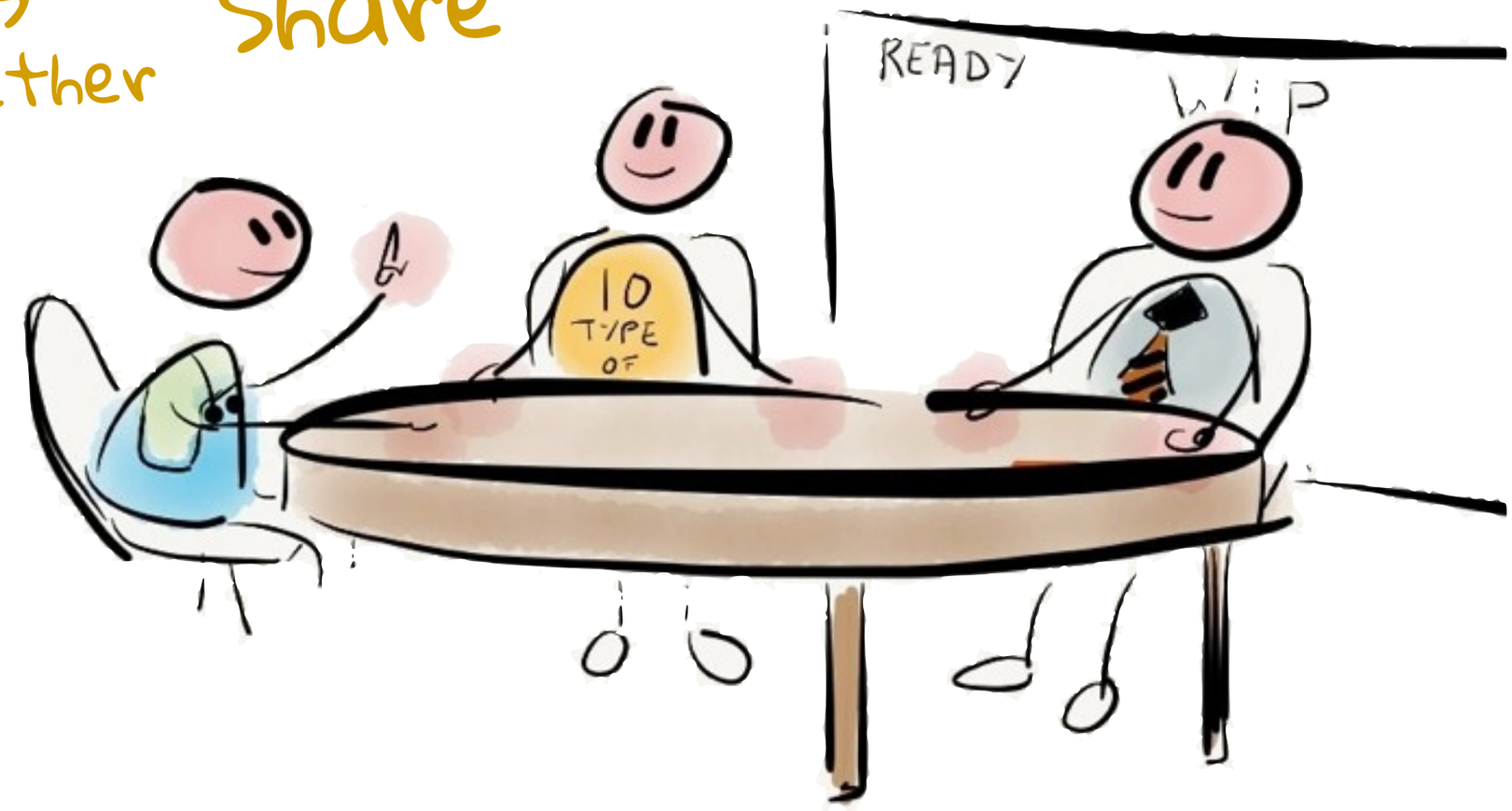
Stand for

Living Documentation

Discuss

Together

Share



Clarify

Examples

Scenario

Elaborate

Requirements

Verify

Can become

Executable

Automation

Stand for

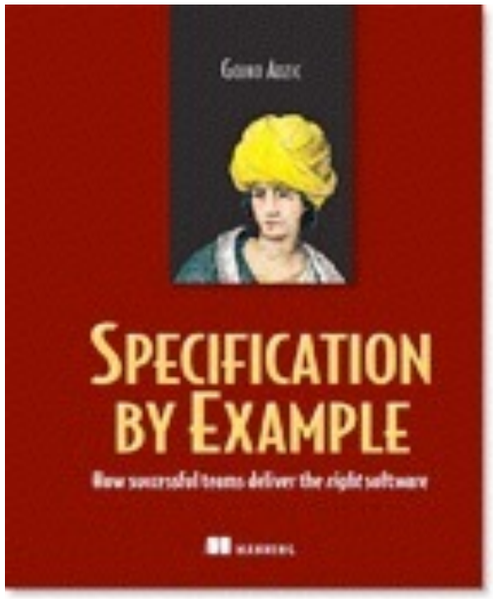
Living Documentation

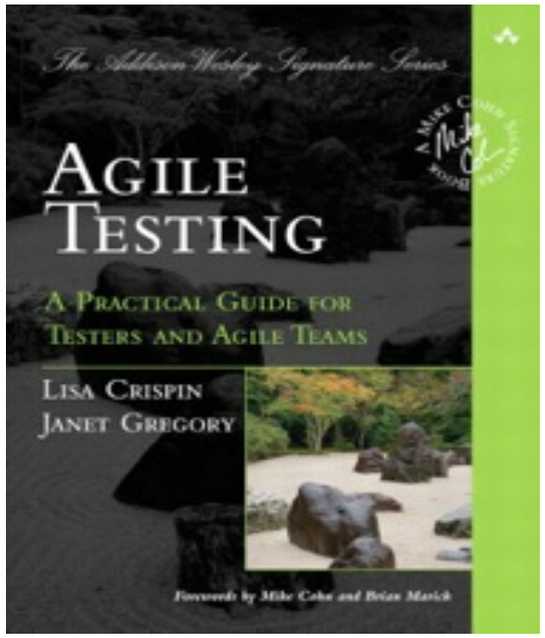
Non Regression Tests

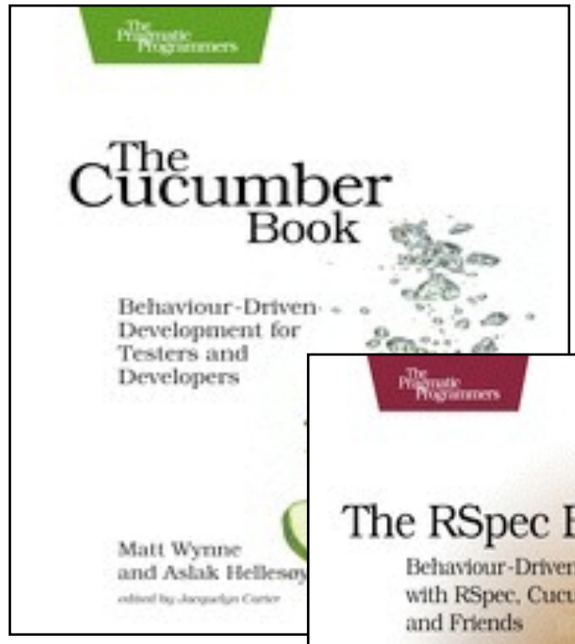
Become

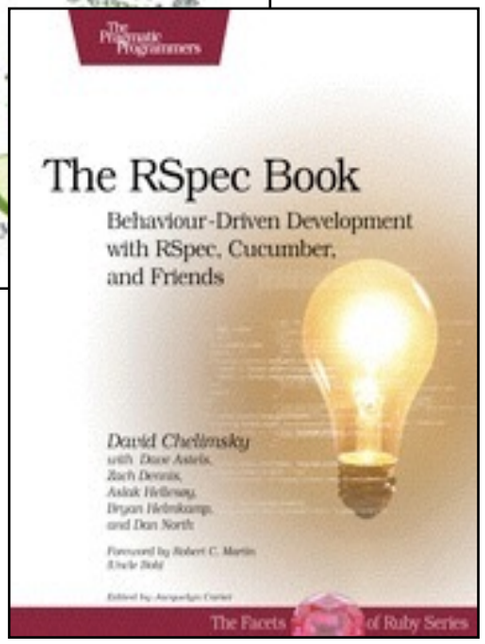
BOOKS

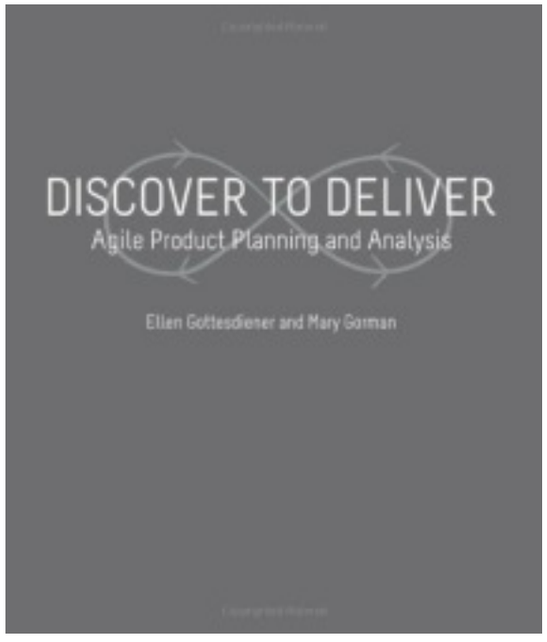
TO GO FURTHER

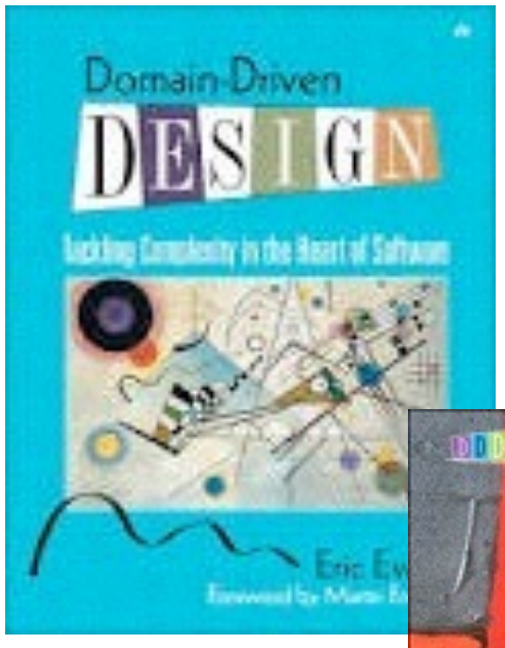


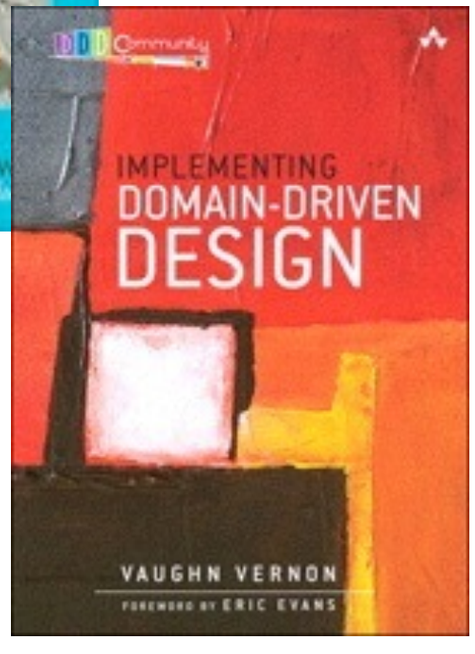












BABOK Agile Extension

IIBA International Institute of Business Analysis

THE POINT

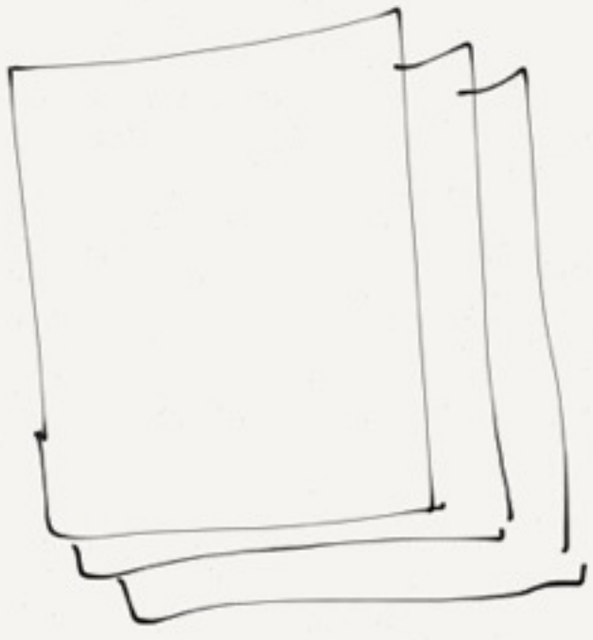
IF ONE THING MUST REMAIN

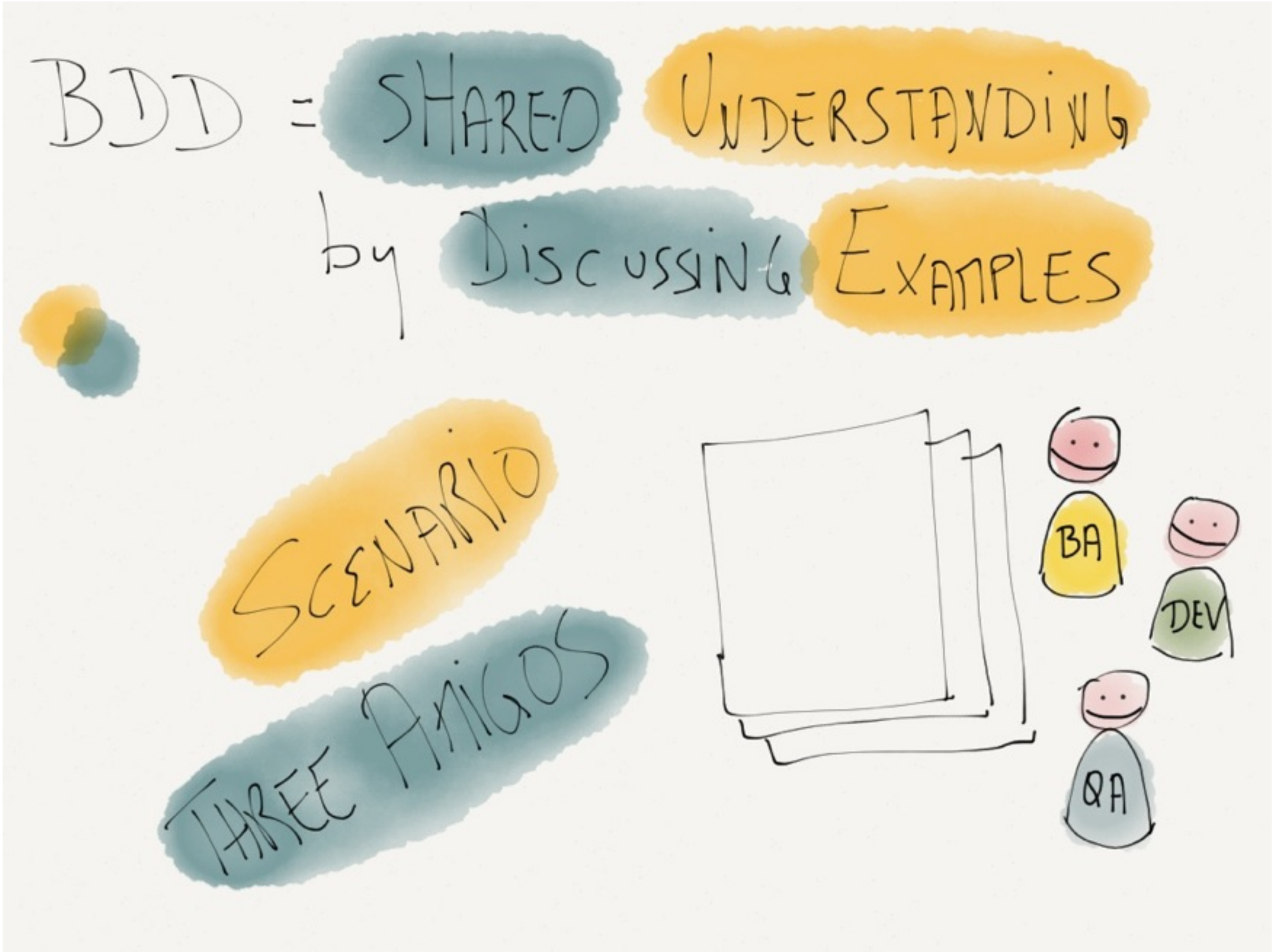
BDD = SHARED UNDERSTANDING
by DISCUSSING EXAMPLES

BDD = SHARED UNDERSTANDING

by DISCUSSING EXAMPLES

SCENARIO



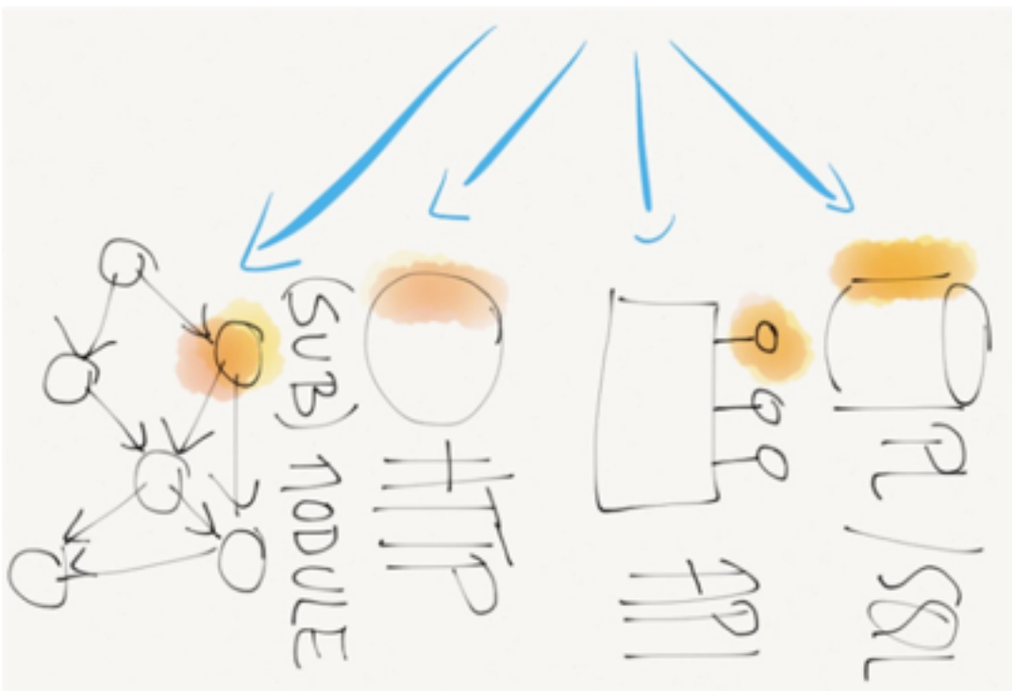

BDD = SHARED UNDERSTANDING
by DISCUSSING EXAMPLES
SCENARIO
THREE AMIGOS
BA
DEV
QA

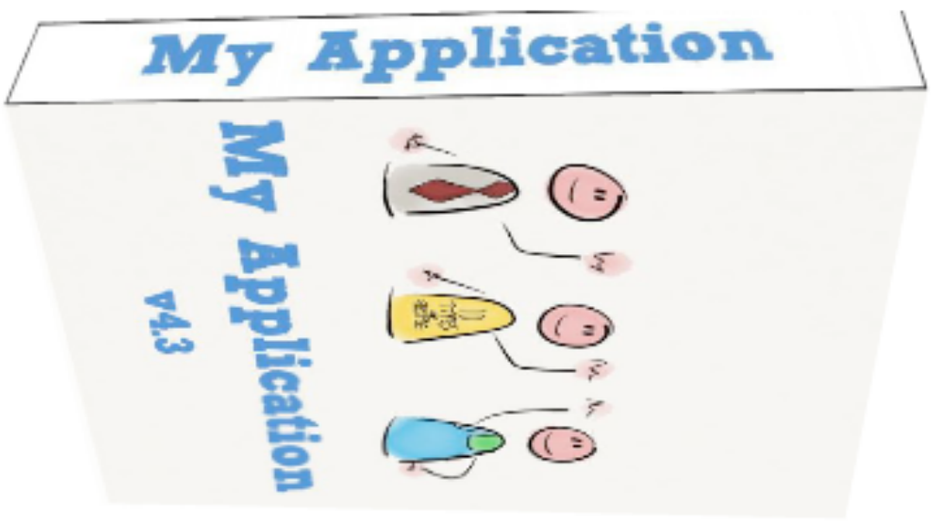
SCENARIO

BDD FRAMEWORK

GLUE CODE

APPLICATION





Stakeholder

Testers

Stakeholders define objectives

Functional and Technical Requirements are defined

Business Requirements are defined

Testers (Quality Assurance) write requirements test plan

Application is tested

Application is deployed in Production

Users use the Application

Developers design and interpret specification into code

Developers write implementation tests

Domain Experts

Developers

At least "6" Sources of Errors among "4" different Roles

Stakeholders define objectives

Business Requirements are defined

Functional and Technical Requirements are defined

Testers (Quality Assurance) write requirements test plan

Developers design and interpret specification into code

Developers write implementation tests

Application is tested

Application is deployed in Production

Users use the Application