

Bordeaux, 22-24 October 2019



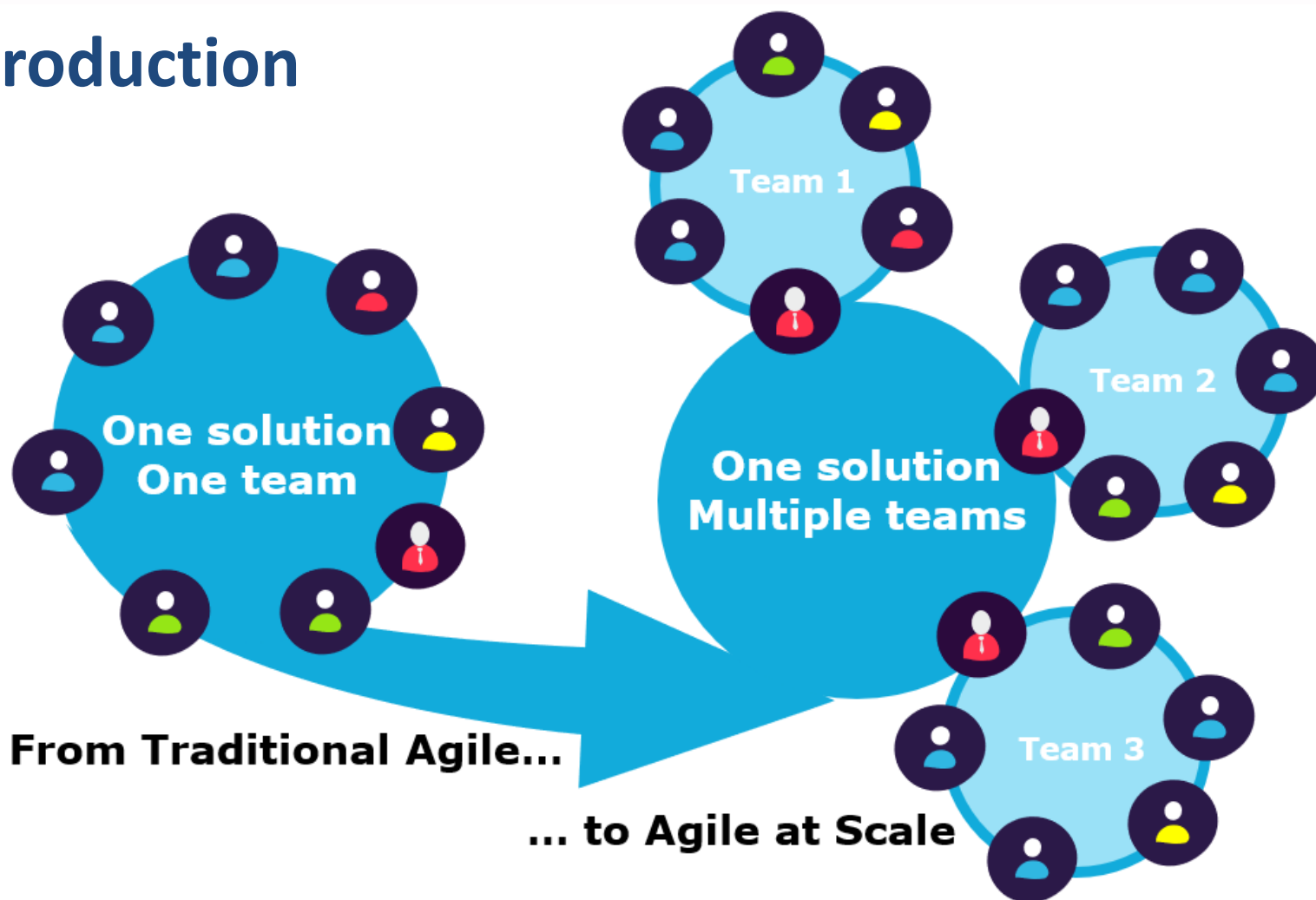
# TESTING IN SAFE<sup>®</sup> : COORDINATE AND OPTIMIZE TEST EFFORTS WITH VISUAL ATDD - EXPERIENCE REPORT AT ORANGE

Presented by Elodie Bernard

# SUMMARY

- Introduction
- What does SAFe® provide to build the quality of a system?
- Challenges for test activities in SAFe®
- Coordinate and optimize testing efforts through visual ATDD and BDD at Orange
- Overview and conclusion

# Introduction





**Bordeaux, 22-24 October 2019**



# WHAT SAFE<sup>®</sup> OFFERS TO BUILD THE QUALITY OF A SYSTEM?

# Building quality with SAFe®

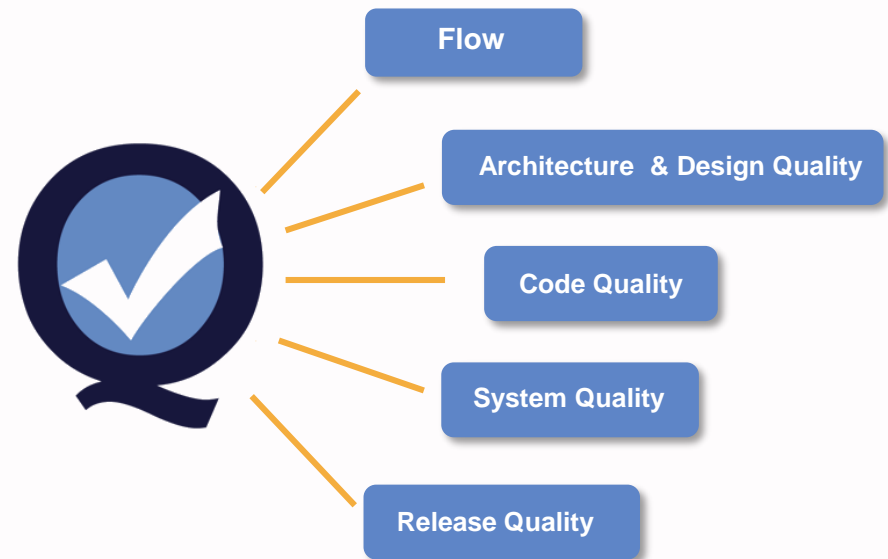
## *Key principles*

- **Key practices:**

- Flow
- Architecture and design quality
- Code quality
- System quality
- Release quality

- **How to integrate them?**

- Via the process definition
- The BDD (Behavior-Driven Development)
- A complete overview of Agile tests



# Building quality with SAFe®

*Create a continuous and efficient workflow with the BDD*

Behavior-driving development (BDD) is a collaborative process that creates a common understanding of requirements between the business and the development team.

## Example of BDD

• Datatable

#	Dataset name	amount	ending_balance	card_type	
1	withdraw \$100	\$100	\$0	Visa	✕ ↓
2	withdraw \$50	\$50	\$50	Visa	✕ ↑ ↓
3	withdraw \$20	\$20	\$80	Mastercard	✕ ↑

⊖ Add new dataset

Edit | View tests

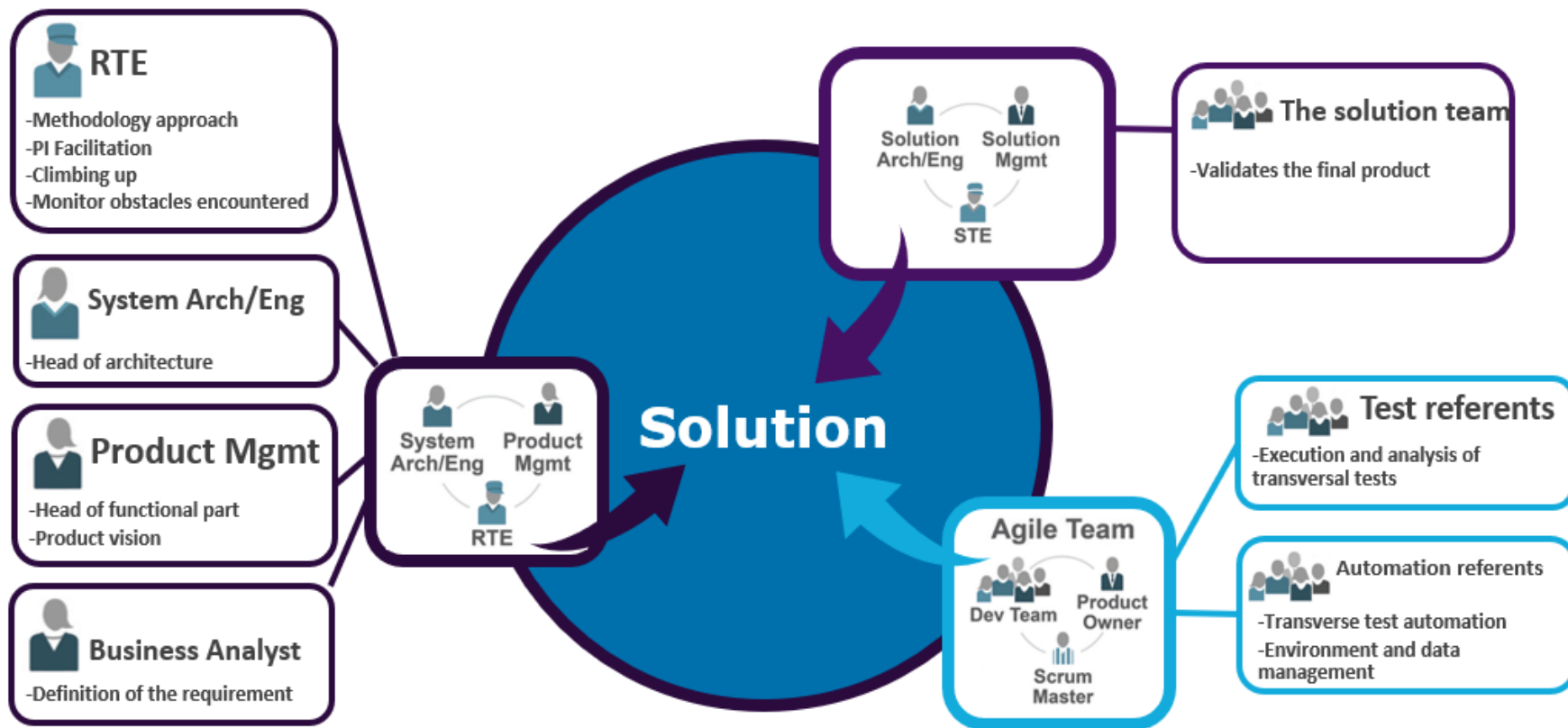
Raw version

- Given the account balance is \$100
- and we have a valide account
- and we introduce a card \${card\_type}
- When the Account Holder requests \${amount}
- and the ATM should dispense \${amount}
- Then the account balance should be \${ending\_balance};
- and the card should be returned

Add step

# Building quality with SAFe®

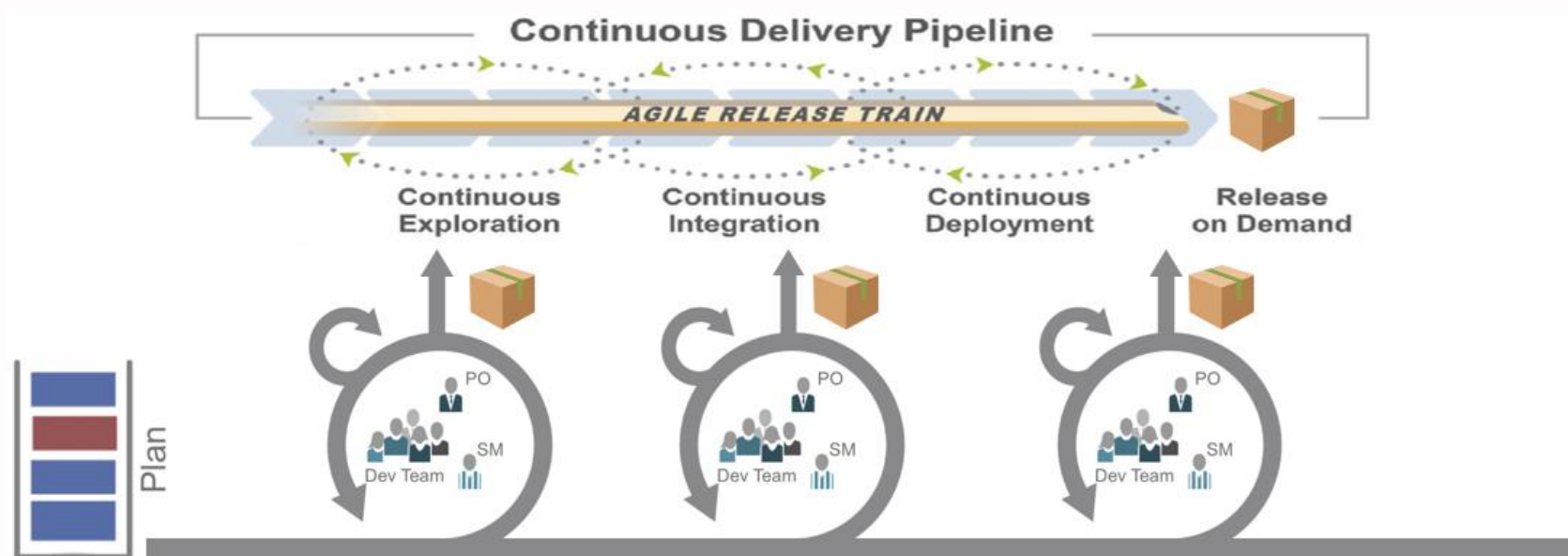
## *The team*



# Building quality with SAFe®

## *Mastering Agile teams with Scrum and Kanban*

- Agile teams are self-organized.
- They define, build, test, deploy increments of the final product.

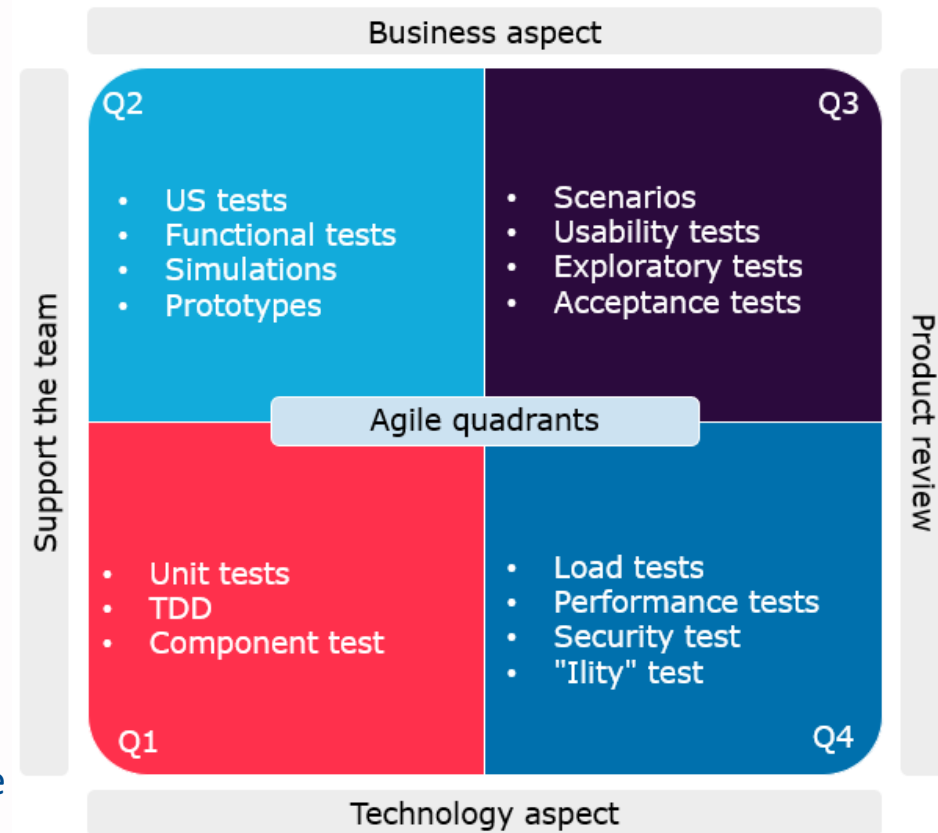




# Building quality with SAFe®

## *Agile testing*

- The use of these four quadrants (Q1 to Q4) promotes the development of a comprehensive strategy that contributes to quality assurance.
- Quadrants Q1 and Q2 contain respectively articles TDD and BDD of SAFe.
- Continuous integration provides more information on the Q3 quadrant.
- Non-functional requirements, continuous integration, and on-demand versions provide more information on the Q4 quadrant.



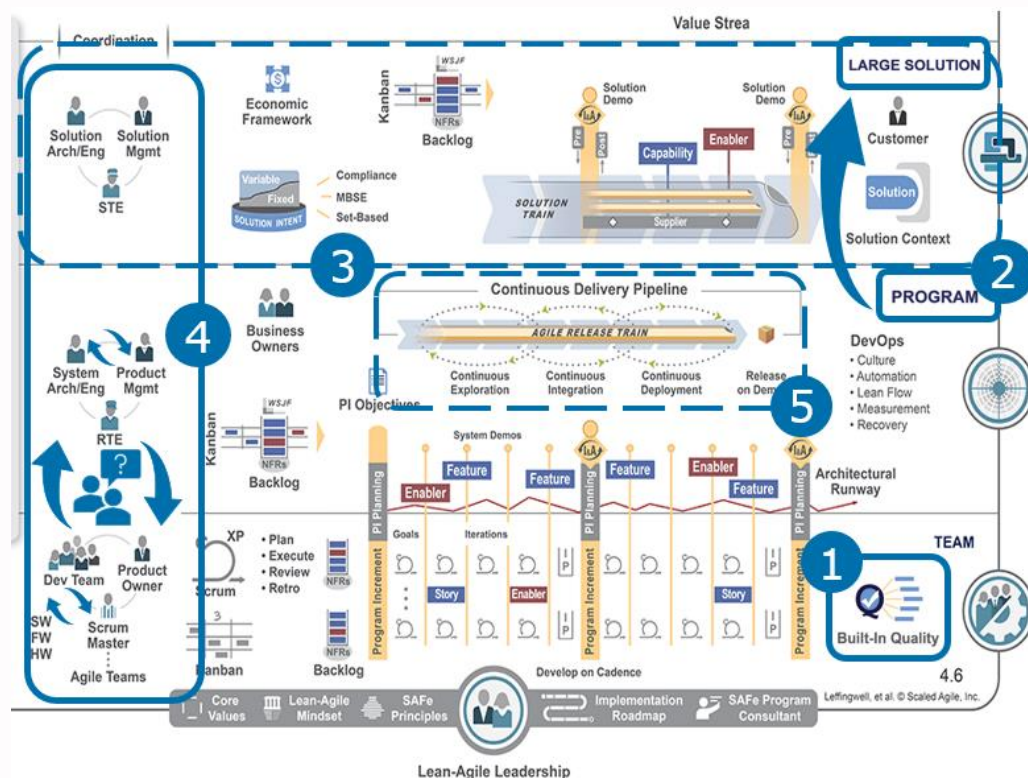
**Bordeaux, 22-24 October 2019**



# CHALLENGES FOR TEST ACTIVITIES IN SAFE<sup>®</sup>

# Challenges for test activities in SAFe®

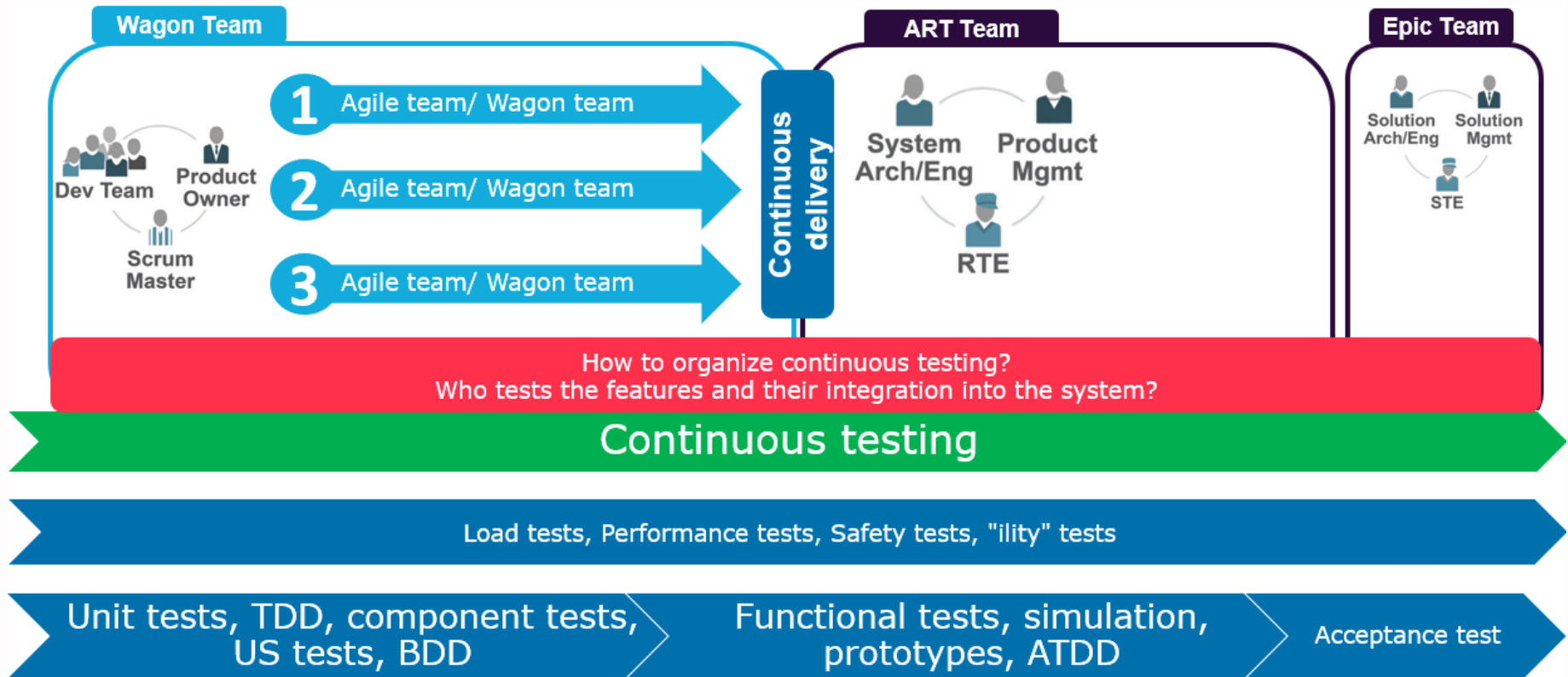
*Activities and roles at the test level are not well defined in SAFe®*



- 1 Test activities not well defined
- 2 Switch from an application QA vision to a solution QA
- 3 Need to secure the right vision of the solution
- 4 Need to coordinate test levels between teams
- 5 Importance of train-level tests to ensure the integration of wagons on the train

# Challenges for test activities in SAFe®

## *Challenges of the testing strategy in SAFe®*





Bordeaux, 22-24 October 2019



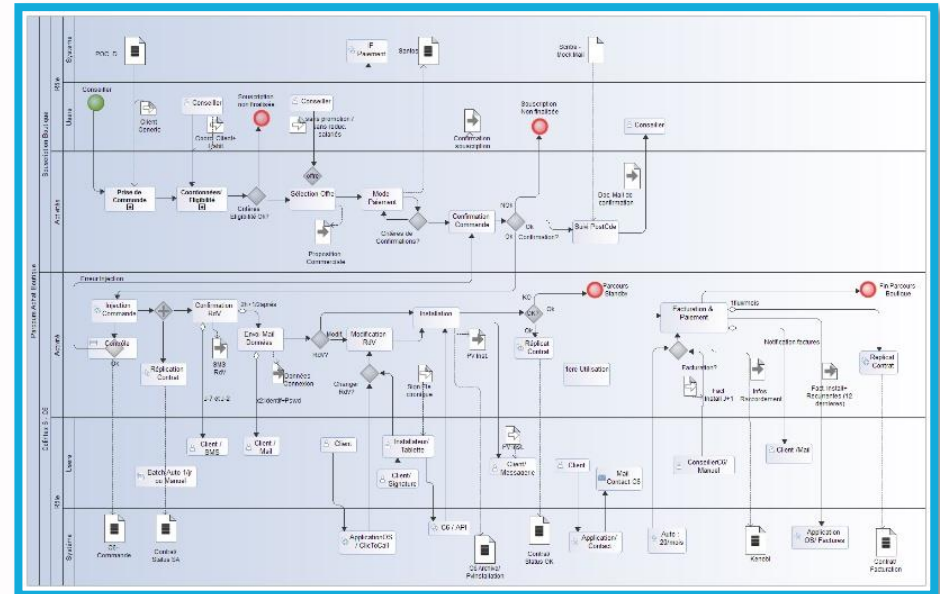
# COORDINATE AND OPTIMIZE TESTING EFFORTS THROUGH VISUAL ATDD AND BDD

ORANGE EXPERIENCE

# From BPMN to visual ATDD

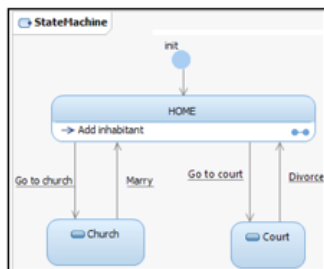
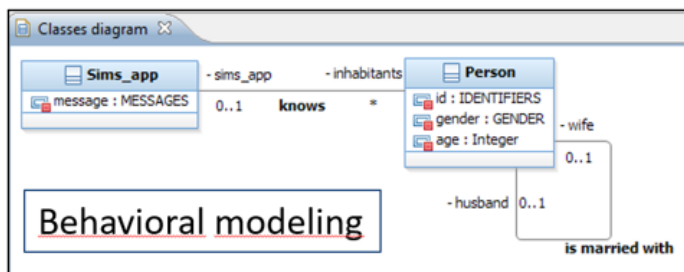
## Constat

- BPMN(Business Process Model and Notation) is initially used
- These models represent the different features of the system and their interaction
- **The strong point of this technique :**
  - made it possible to clarify the need
- **The weaks points of this technique :**
  - it does not allow to vary the level of abstraction easily
  - not make it possible to include testing phase
  - does not allow communication adapted to each level of the train



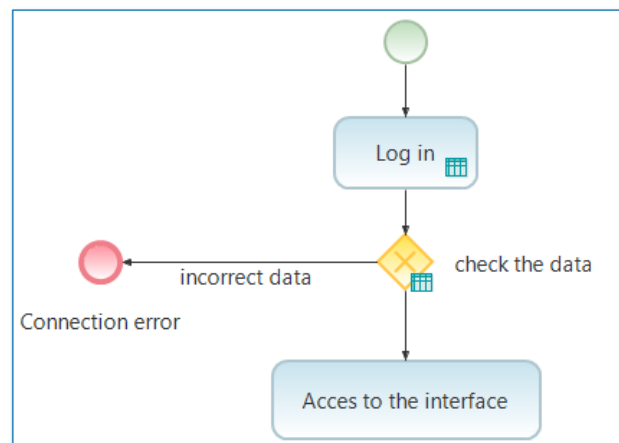
# From BPMN to visual ATDD

## Visual ATDD ?



```

Mary II
---REQ: MARRIAGE
if person1 = person2 then
    ---BAIN: Error same person
    self.message = MESSAGES::ERROR_MARRIAGE_SAME_PERSON
else if person1.gender = person2.gender then
    ---BAIN: Error same gender
    self.message = MESSAGES::ERROR_MARRIAGE_SAME_GENDER
else if (person1.age < 18) or (person2.age < 18) then
    ---BAIN: Error one is not adult
    self.message = MESSAGES::ERROR_MARRIAGE_NOT_AN_ADULT
else if person1.is_married() or person2.is_married() then
    ---BAIN: Error one is already married
    self.message = MESSAGES::ERROR_MARRIAGE_NOT_SINGLE
else
    ---BAIN: Success
    self.message = MESSAGES::NO_MESSAGE
    and
    if person1.gender = GENDER::MALE then
        ---BAIN: person1 is a man
        person1.wife = person2
    else
        ---BAIN: person1 is a woman
        person1.husband = person2
    endif
endif
    
```

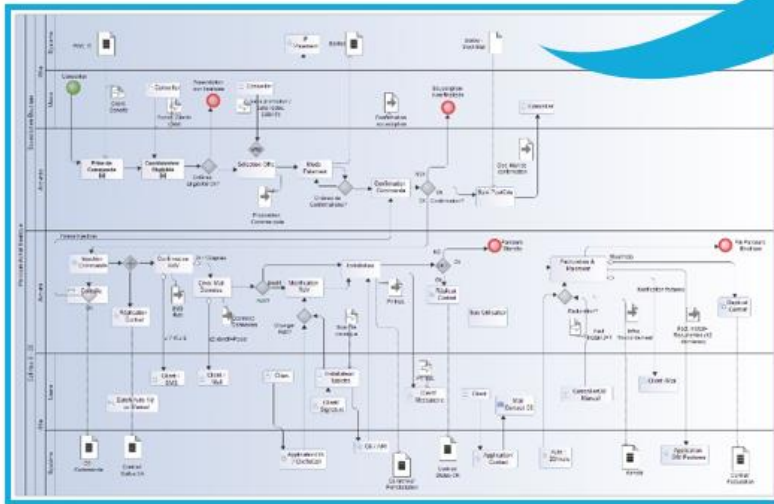
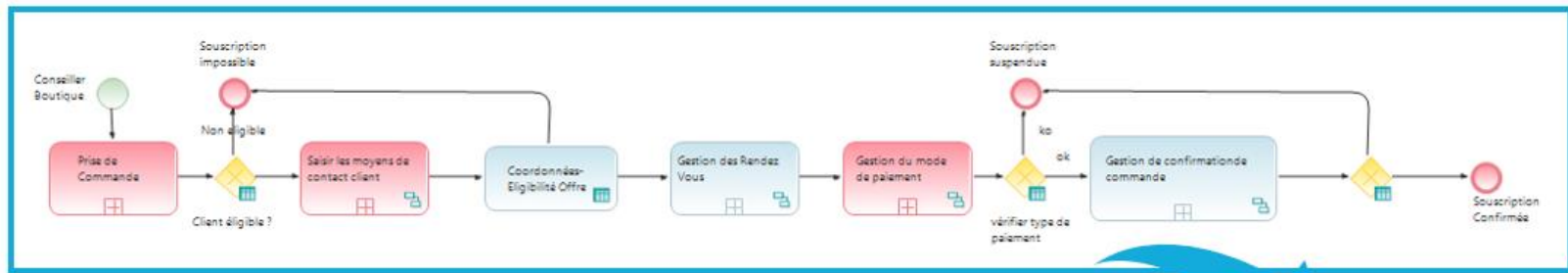


	id	password	Test steps	Outcome
1	correct	correct	1 Check the data	The data are correct → Access to the application
2	correct	incorrect	1 Check the data	The data are incorrect check the data → incorrect data → Connection error
3	incorrect	incorrect	1 Check the data	The data are incorrect check the data → incorrect data → Connection error

From traditional MBT practices to a lean approach: **the visual ATDD**

# From BPMN to visual ATDD

## Solution



10 scénarios de test dans le dossier PACHONEH-version P1#4

4

6

3 indicateurs de couverture

features 3

2

1

Objectif 10

5

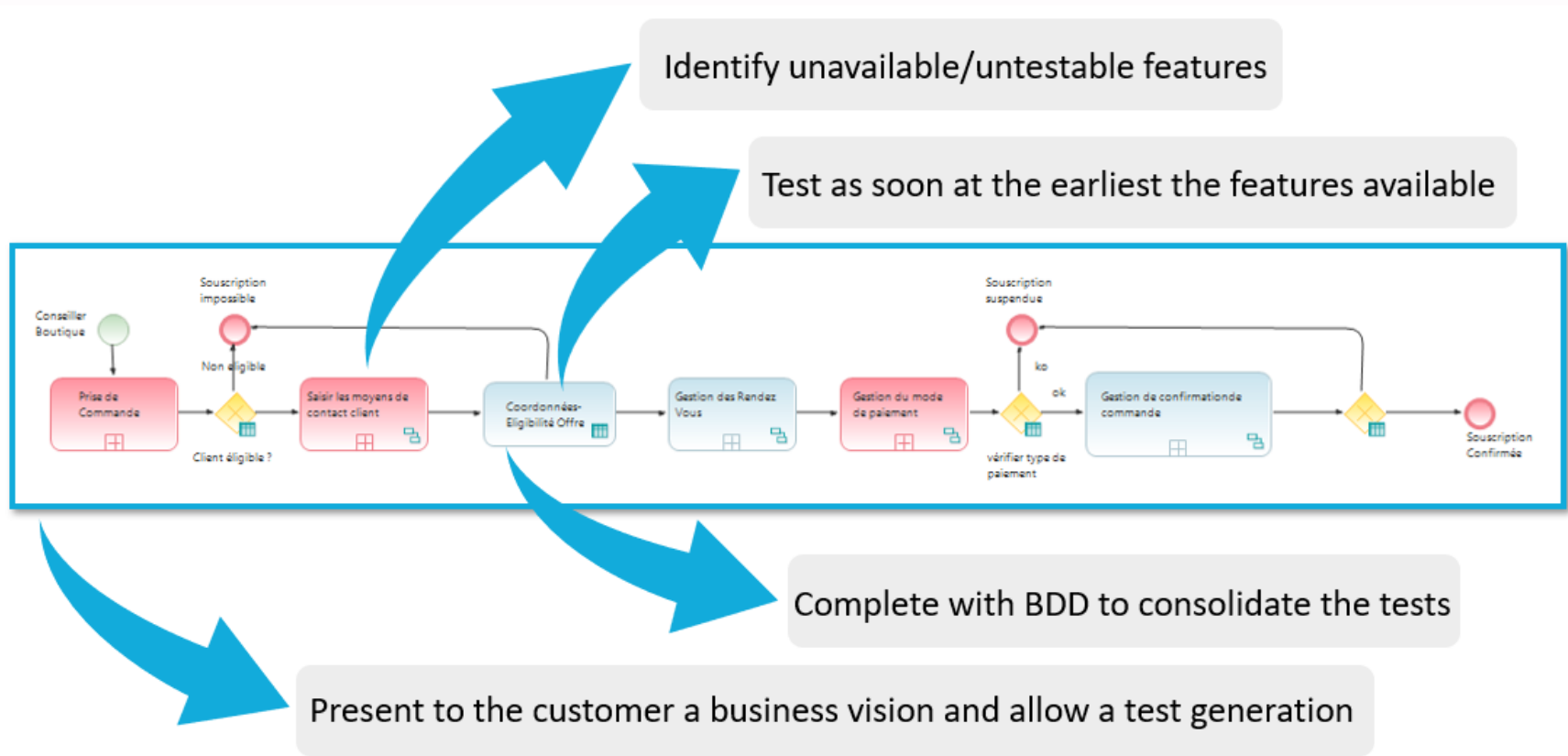
5

✓ Choisir les RdV 0/1 (0%)	PACHONEH-version P1#4 / SousProcessus_PAB
✓ Gestion de confirmation de commande 0/2 (0%)	PACHONEH-version P1#4 / SousProcessus_PAB
✓ Gestion de l'injection C6 1/2 (50%)	PACHONEH-version P1#4 / SousProcessus_PAB
✓ Gestion du mail confirmation 1/2 (50%)	PACHONEH-version P1#4 / SousProcessus_PAB
✓ Gestion du mode de paiement 0/4 (0%)	PACHONEH-version P1#4 / SousProcessus_PAB
✓ PAB_Souscription_Offre_Boutique 10/26 (38%)	PACHONEH-version P1#4 / Parcours génération
✓ Saisir les moyens de contact client 1/2 (100%)	PACHONEH-version P1#4 / SousProcessus_PAB
✓ Souscription Commande 4/28 (32%)	PACHONEH-version P1#4 / SousProcessus_PAB



# From BPMN to visual ATDD

*Via ATDD*



# Coordinate and optimize testing efforts through *visual ATDD and BDD*

## BDD

### Datatable

#	Dataset name	amount	ending_balance	card_type	
1	withdraw \$100	\$100	\$0	Visa	⌕ ⬇
2	withdraw \$50	\$50	\$50	Visa	⌕ ⬆ ⬇
3	withdraw \$20	\$20	\$80	Mastercard	⌕ ⬆

⊕ Add new dataset

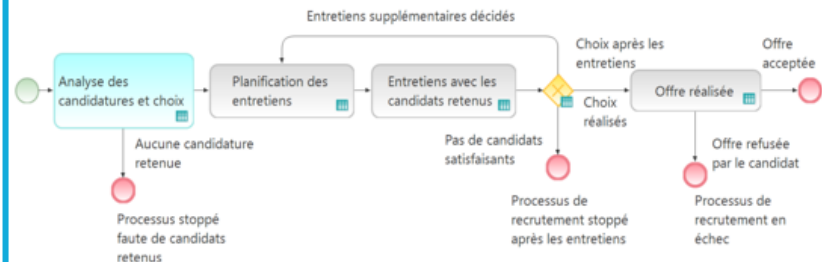
✎ Edit | ⌕ View tests

🔗 Raw version

1	Given the account balance is \$100	⌕ ⌕ ⌕
2	and we have a valide account	⌕ ⌕ ⌕
3	and we introduce a card \$(card_type)	⌕ ⌕ ⌕
4	When the Account Holder requests \$(amount)	⌕ ⌕ ⌕
5	and the ATM should dispense \$(amount)	⌕ ⌕ ⌕
6	Then the account balance should be \$(ending_balance)	⌕ ⌕ ⌕
7	and the card should be returned	⌕ ⌕ ⌕

Add step

## Visual ATDD



CV_Anonyme	Étapes de test	Direction suivante	User Story
1 Non	1 Evaluation CV par les	Check - Evaluations saisies et	→ Planification des entretiens
2 Oui	1 Evaluation CV par les	Check - Evaluations saisies et	→ Planification des entretiens
3 Non	Evaluation CV par les intervenants	Check - Evaluations saisies et	Analyse des candidatures et choix - Aucun

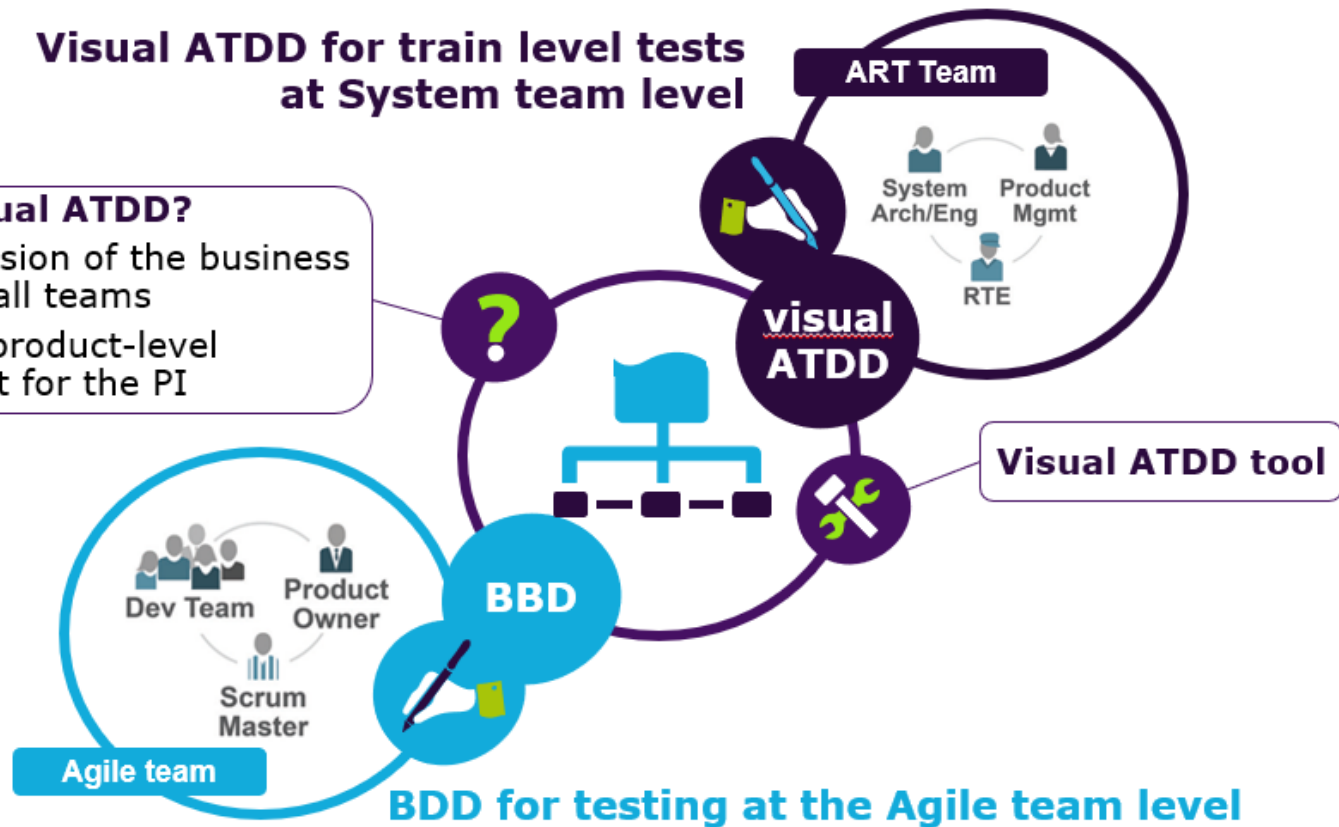
Visual ATDD makes it possible to model business processes and thus strengthen and optimize communications in a SAFe® context

# Coordinate and optimize testing efforts through visual ATDD and BDD

**Visual ATDD for train level tests  
at System team level**

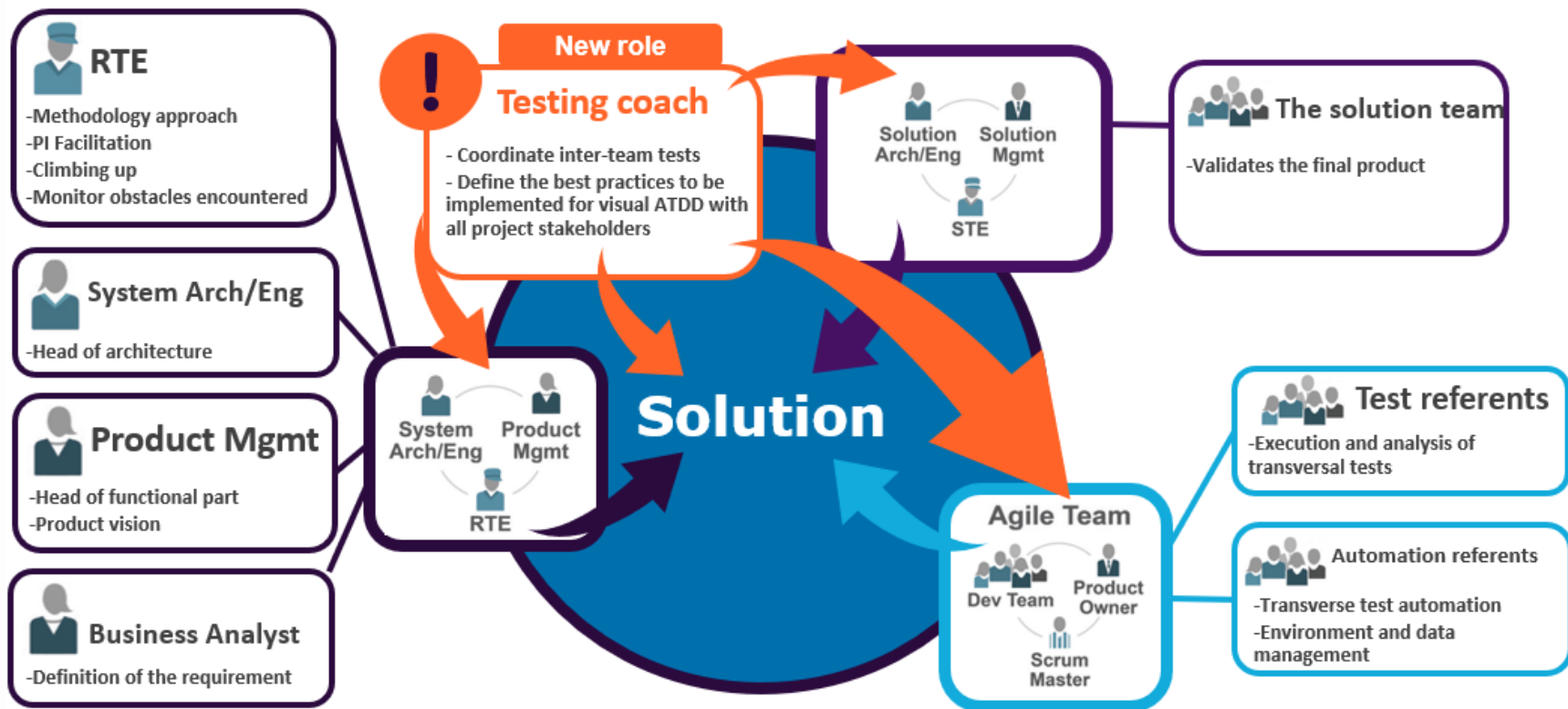
## Why use visual ATDD?

- Provide a vision of the business solution to all teams
- Define the product-level requirement for the PI



*ATDD and BDD two complementary and synergistic approaches*

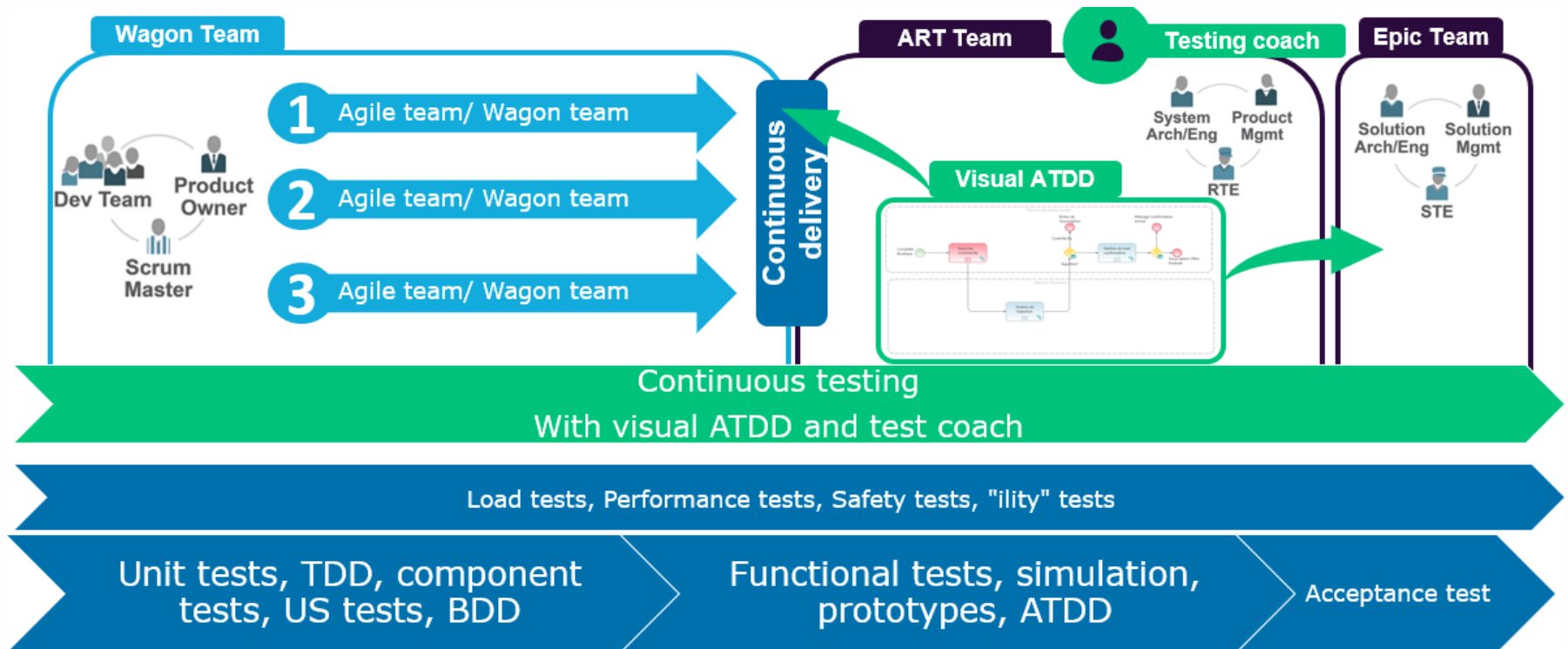
# Coordinate and optimize testing efforts through visual ATDD and BDD



*The testing coach*



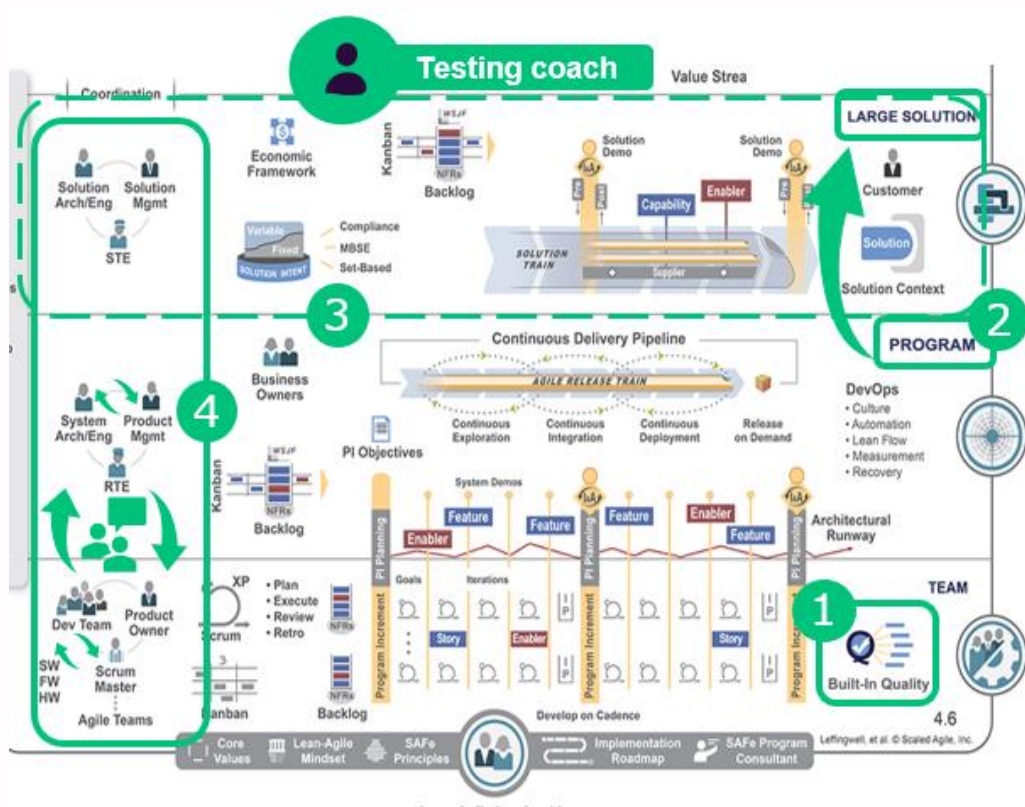
# Coordinate and optimize testing efforts through visual ATDD and BDD



*Test strategy*

# Overview and conclusion

*With visual ATDD :*



- 1 The roles related to the test are better defined through the addition of a "Testing coach" role to coordinate the tests.
- 2 It is easier to switch from an application QA vision to a solution QA with visual ATDD and its different levels of abstraction.
- 3 The vision and communication of the solution are strengthened by shared modeling
- 4 Models can represent the progress of the different test levels according to the teams and thus better coordinate them