Toward testing multiple User Interface versions

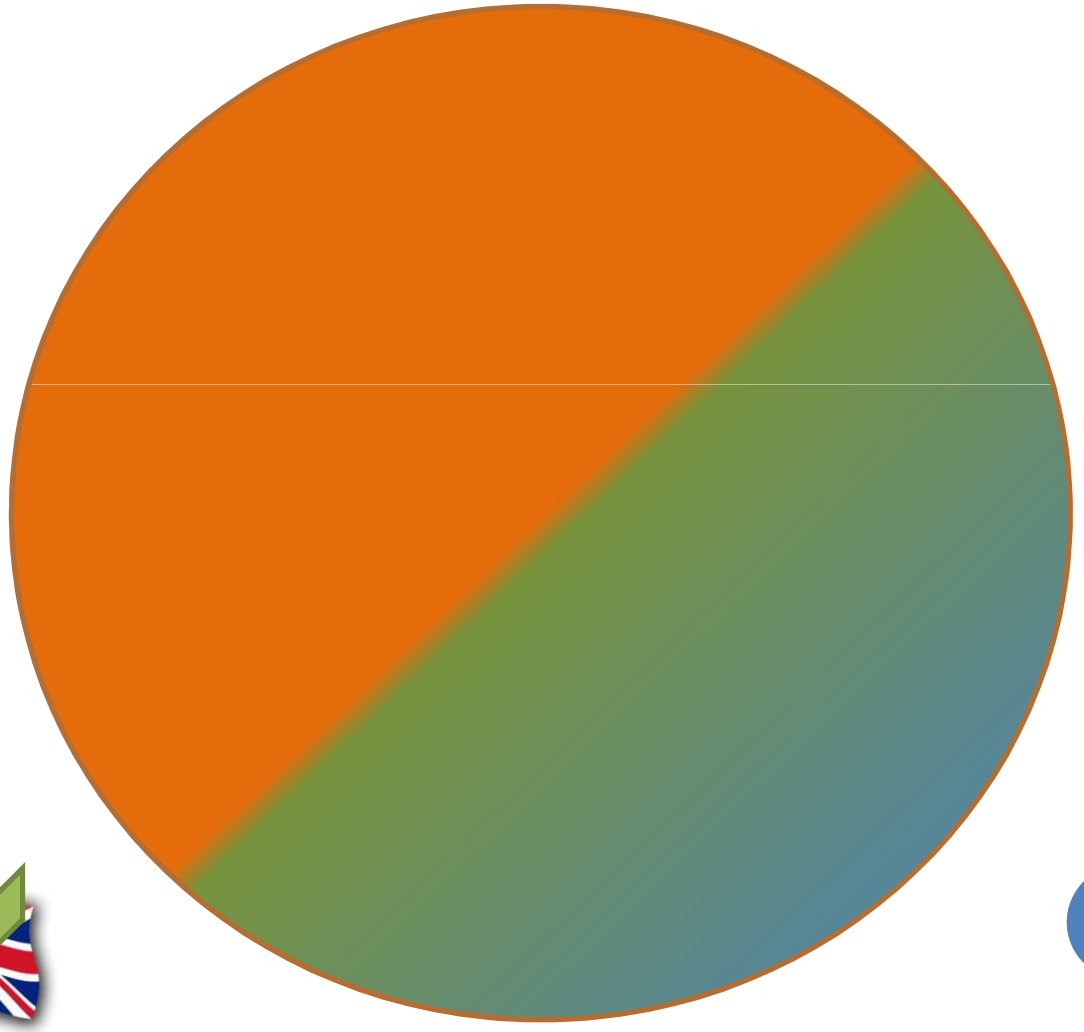
Nelson Mariano Leite Neto,
Julien Lenormand,
Lydie du Bousquet,
Sophie Dupuy-Chessa

Background: Adaptable UI

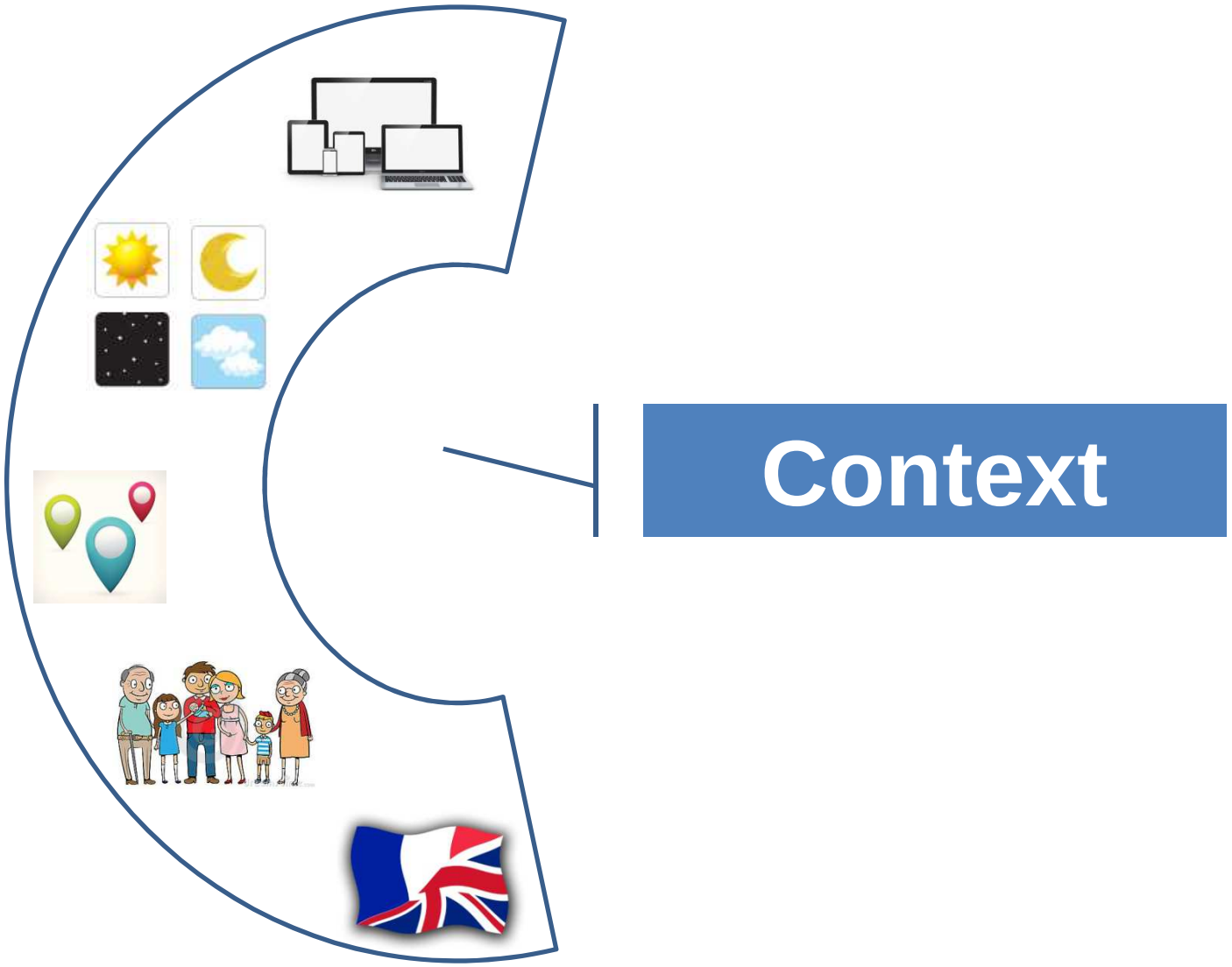
Background: Adaptable UI

Background: Adaptable UI

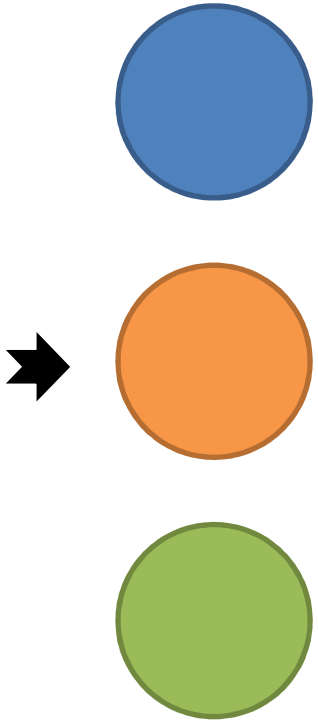
Background: Adaptable UI



Background: Adaptable UI

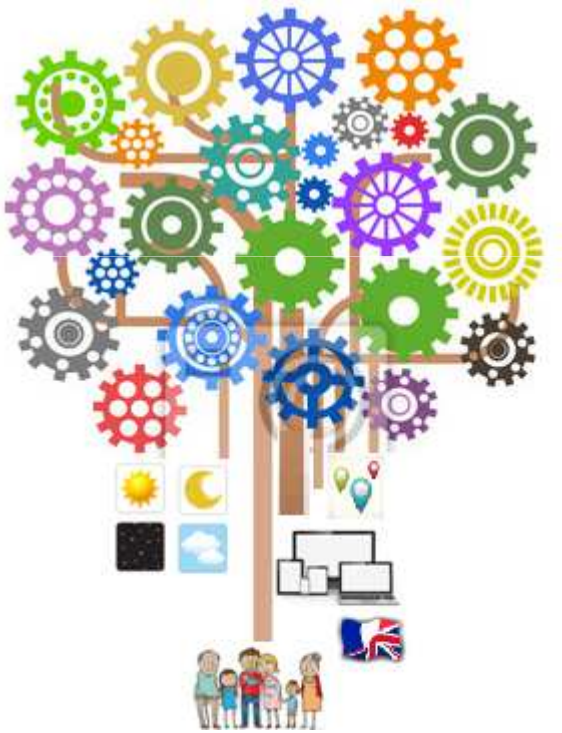


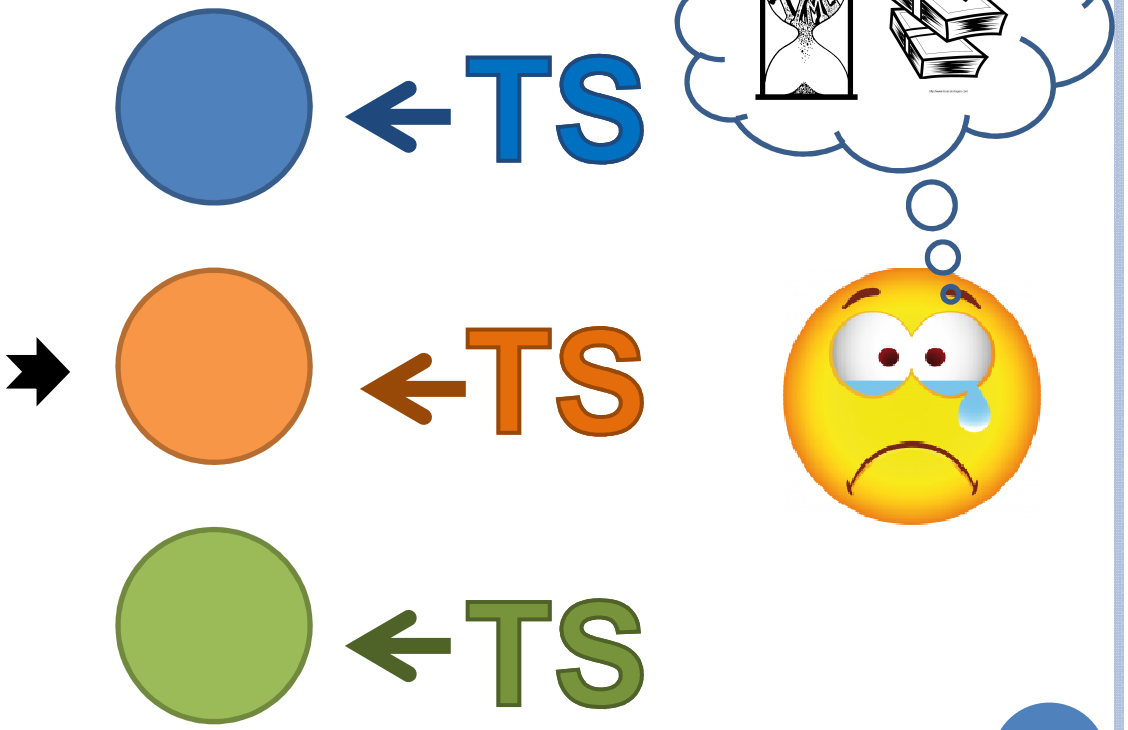
Validation challenges with Adaptable UI



Achieve quality
Cost-effective way

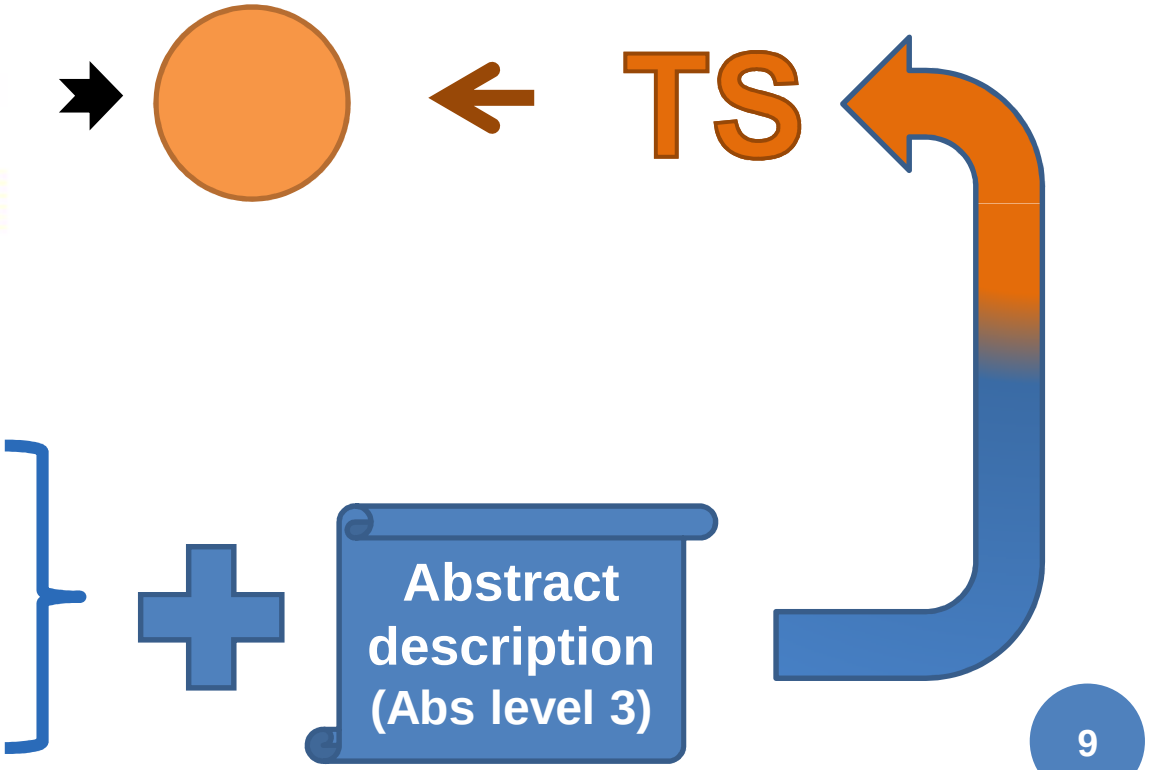
BASIC TESTING APPROACH



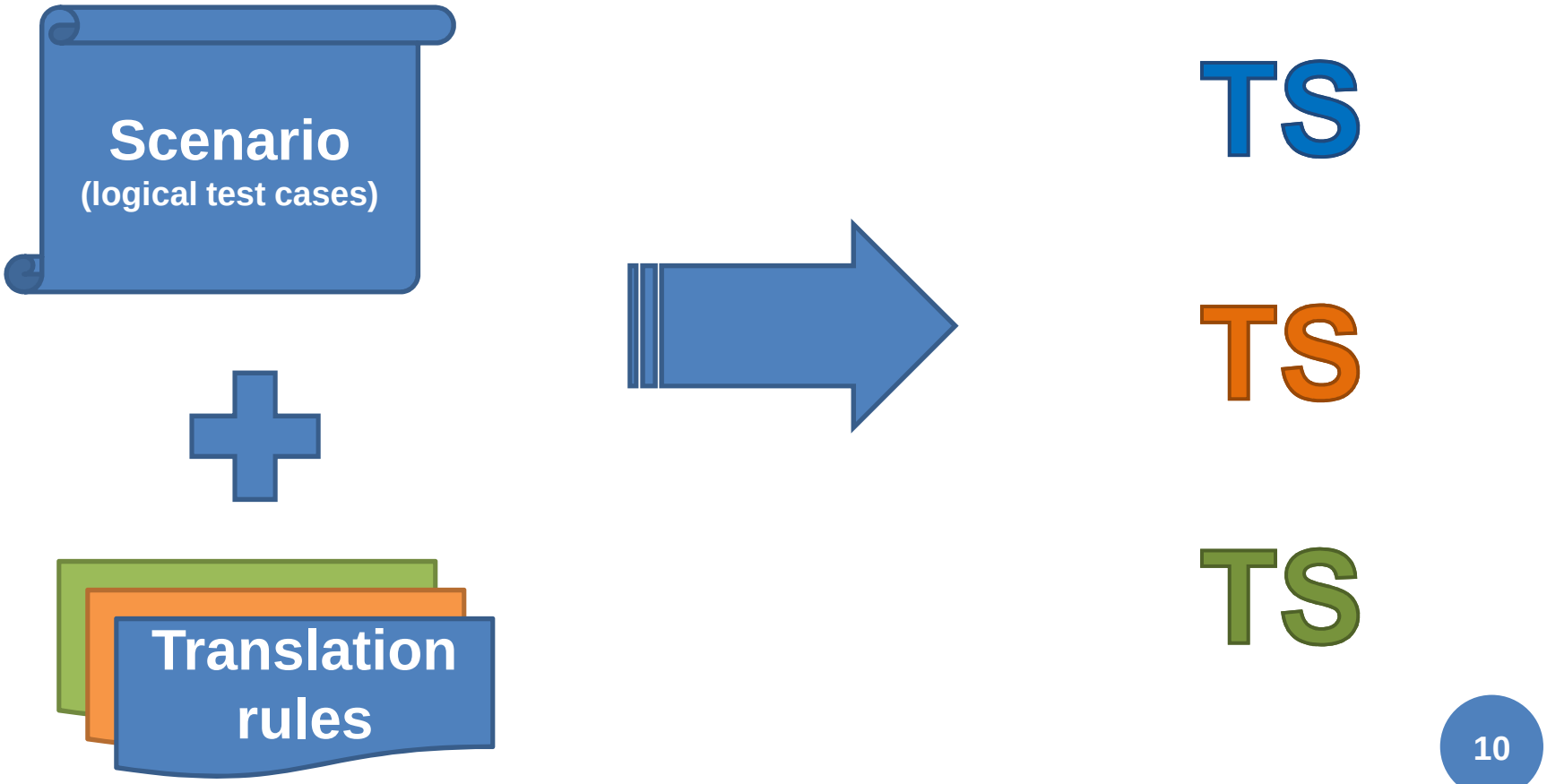


Our final goal

Generate tests w.r.t. the context



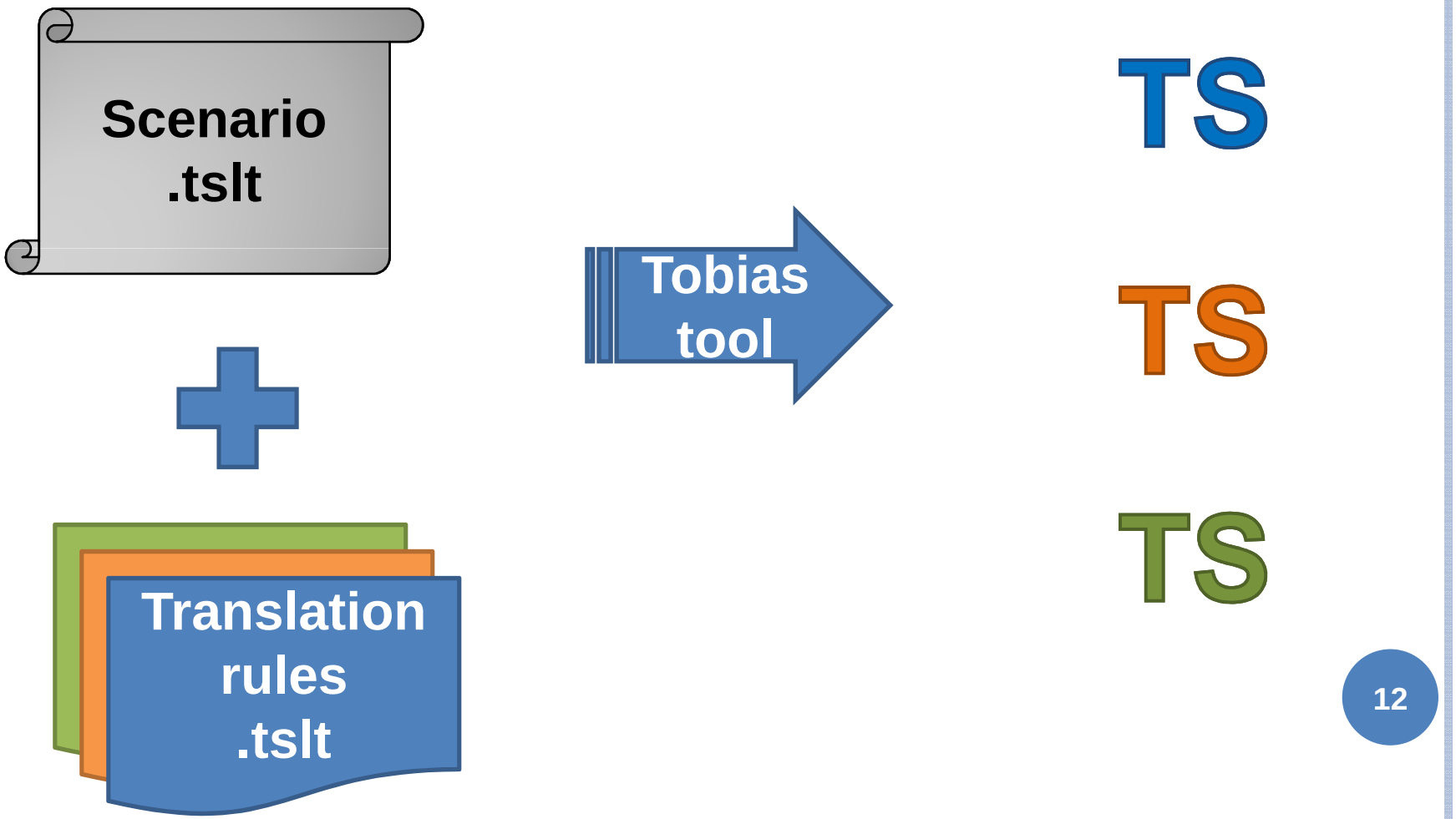
Step 0:
Different test scripts from one scenario



TOWARD TESTING MULTIPLE USER INTERFACE VERSIONS

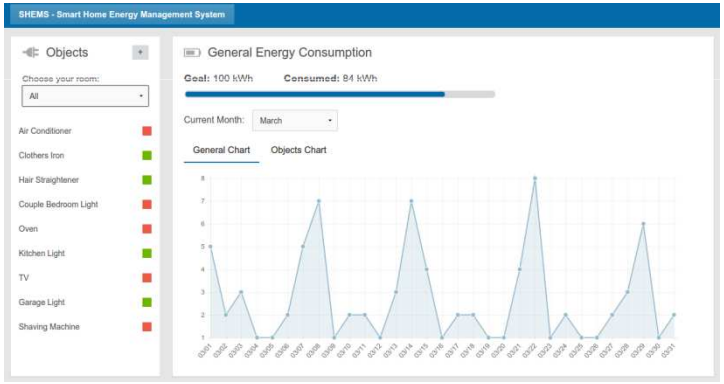
Nelson Mariano Leite Neto,
Julien Lenormand,
Lydie du Bousquet,
Sophie Dupuy-Chessa

Different test scripts from one scenario



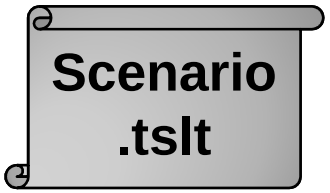
AN EXAMPLE

Smart home energy consumption monitoring application



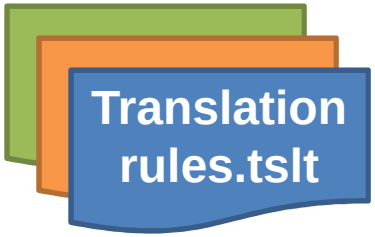
4 different interfaces (**Mobile**, **Web0**, **Web1**, **Web2**)

An abstract description as a testing scenario




```
group testMonthValue[us=true] {
    Integer month = [1-3];
    @goToGoal;
    @selectMonth;
    @verifyValues;
}
```

SPECIALIZED TRANSLATION RULES



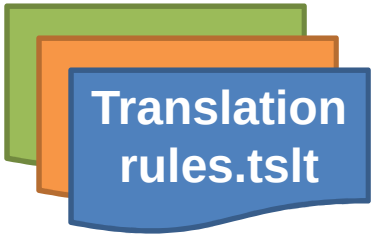
Mobile	group goToGoal[us=false] {
	// does nothing
	}

Web0	group goToGoal[us=false] {
	driver.get(siteAddress);
	}

Web1	group goToGoal[us=false] {
	driver.get(siteAddress);
	WebElement goalButton =
	driver.findElement(By.xpath("/html/body/div/section/ul/li[1]/div/a"));
	goalButton.click();
	}

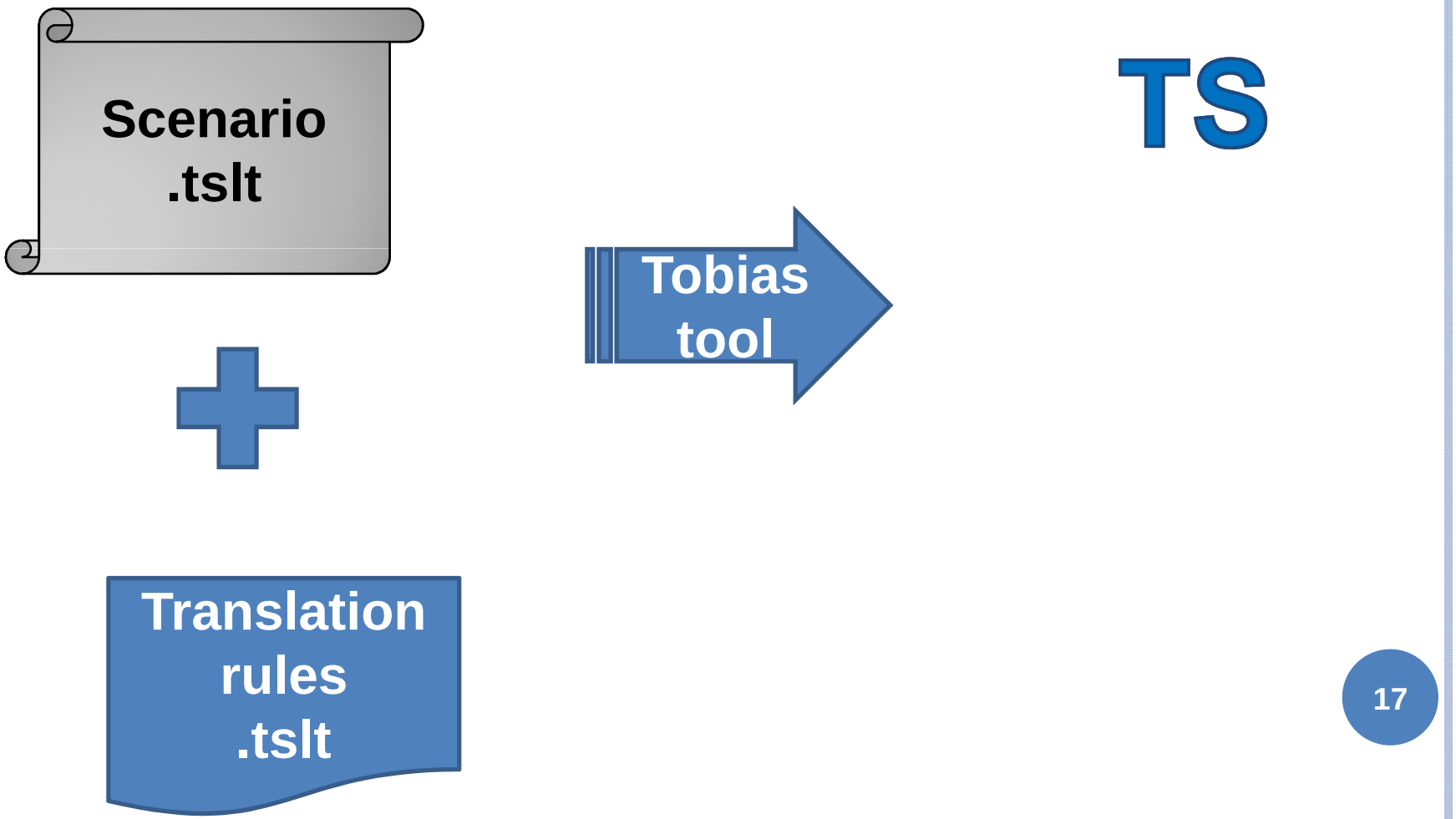
Web2	group goToGoal[us=false] {
	WebElement goalButton =
	driver.findElement(By.id("menu_goal"));
	goalButton.click();
	}

SPECIALIZED TRANSLATION RULES

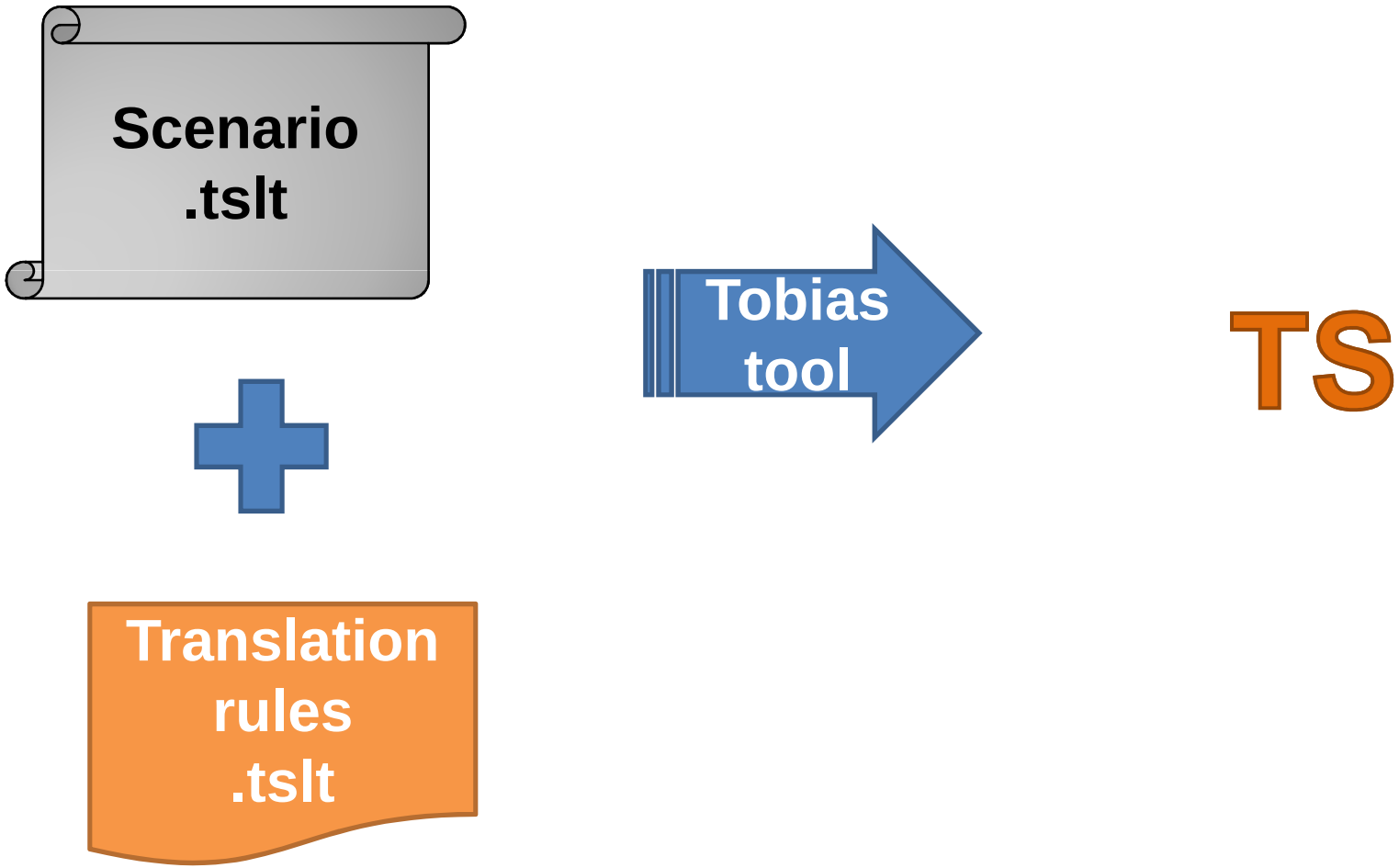


- From asbtract to executable level
- Target the testing tool framework
- Modular
  - Easy to produce
  - Easy to maintain
- Produce manually (step 0)

Different test scripts from one scenario



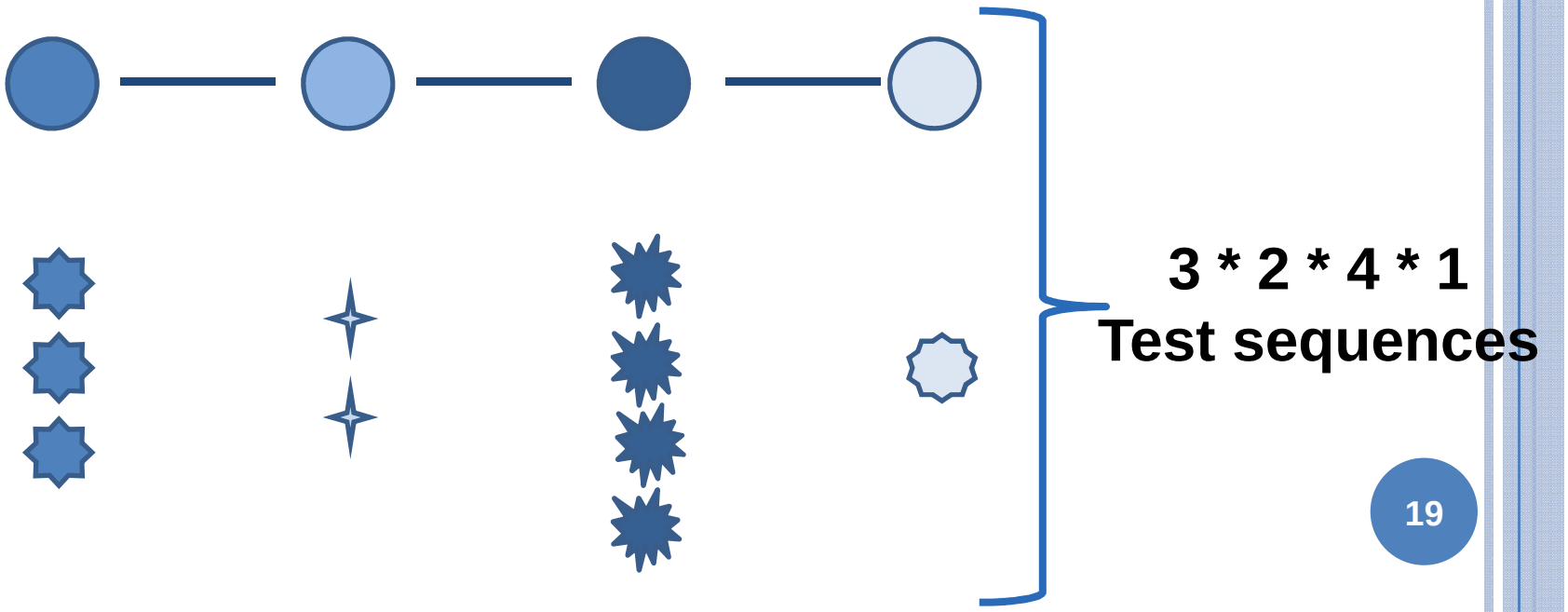
Different test scripts from one scenario



THE ROLE OF TOBIAS COMBINATORIAL UNFOLDING

Tobias tool

Tobias is a combinatorial testing tool
Unfold test suites in a combinatorial way



THE ROLE OF TOBIAS


```
group testMonthValue[us=true] {
        Integer month = [1-3];
...                              }
```

Means 3 different test scripts
from a single description

Some results: Translation rule size

Feature	Rule	Mobile	Web0	Web1	Web2
Goal	@goToGoal	0	0	3	2
	@selectMonth	5	3	3	3
	@verifyValues	4	4	4	4
	@veryfyMonthsCount	6	3	3	3
Filter	@goToObjects	2	0	3	2
	@selectRoom	3	5	5	5
	@verifyObjectsFiltered	25	5	5	5
	@selectRoomUncorrect	3	3	3	3
Compare	@goToCompare	2	3	3	2
	@selectRoom	3	3	3	3
	@verifyWidget	10	10	10	10
	@verifyChart	0	0	0	0
	Total	63	39	45	42

Some results:
Test scripts size

Feature tested	Mobile	Web0	Web1	Web2
Goal	161	143	151	147
Filter	529	350	402	369
Compare	231	235	235	227

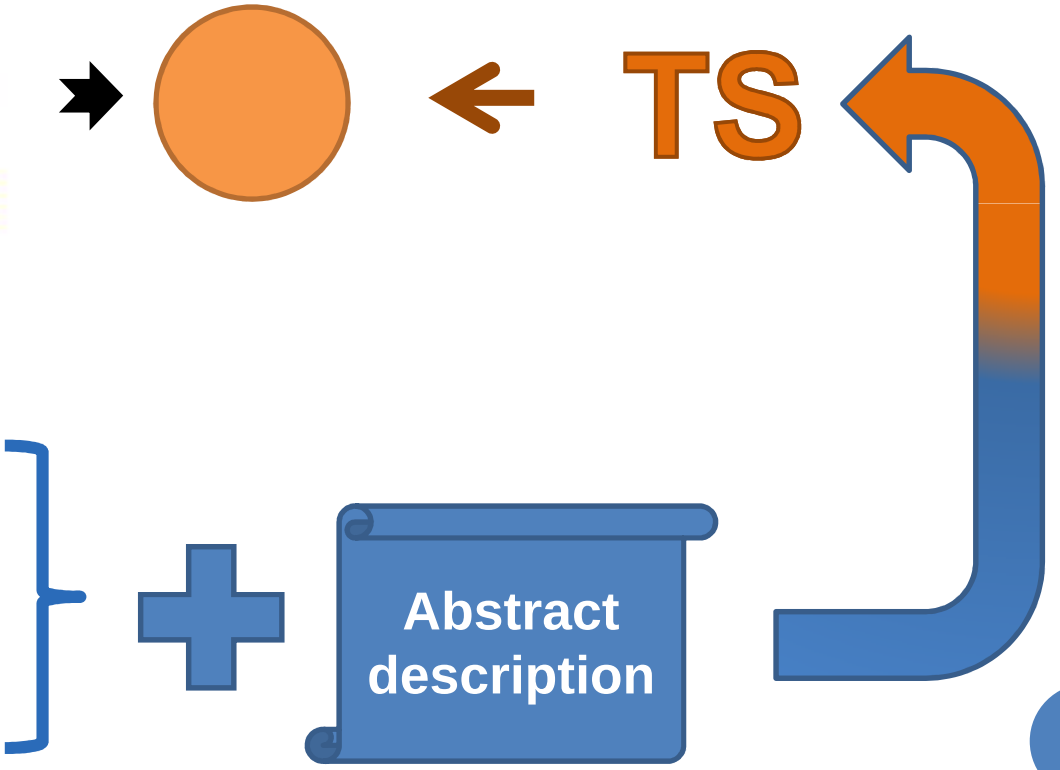
STEP 0

CONCLUSIONS

From one abstract description
to specialized concrete tests

- Possible
- Relevant
- Cost effective
- Limitations are in **Tobias** and **test drivers**
- Need more experimentations

NEXT STEPS



NEXT STEPS

- Generate automatically abstract scenarios from a model
- Generate automatically translation rules
- Relate context to test generation