

Improving Test Coverage and Maintainability



Emily Bache

Technical Coach

@emilybache | coding-is-like-cooking.info

Summary

Approval Testing

Code Coverage

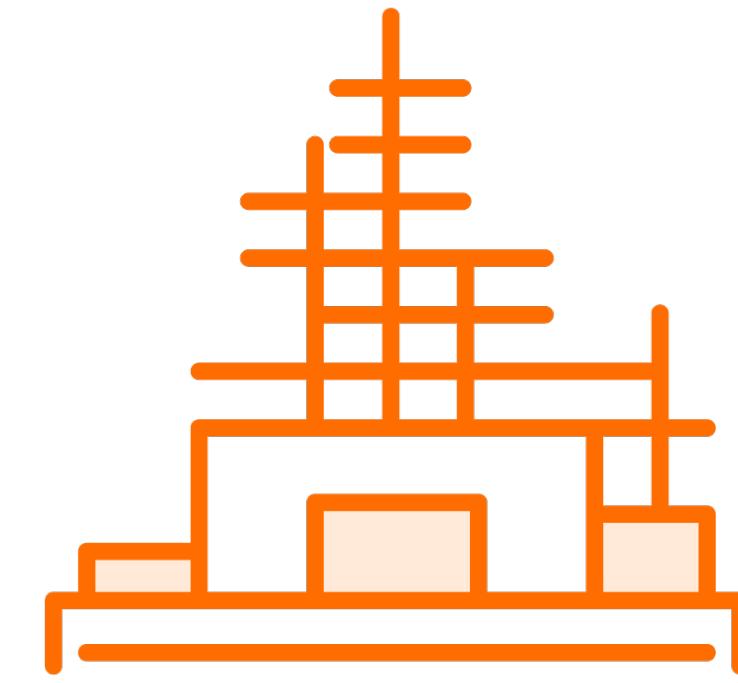
All Combinations

Mutation Testing

Demo - Supermarket Receipt

Replacing an Assertion with Approvals

The Three Parts of a Test



Arrange

Set up the object to be tested, and collaborators

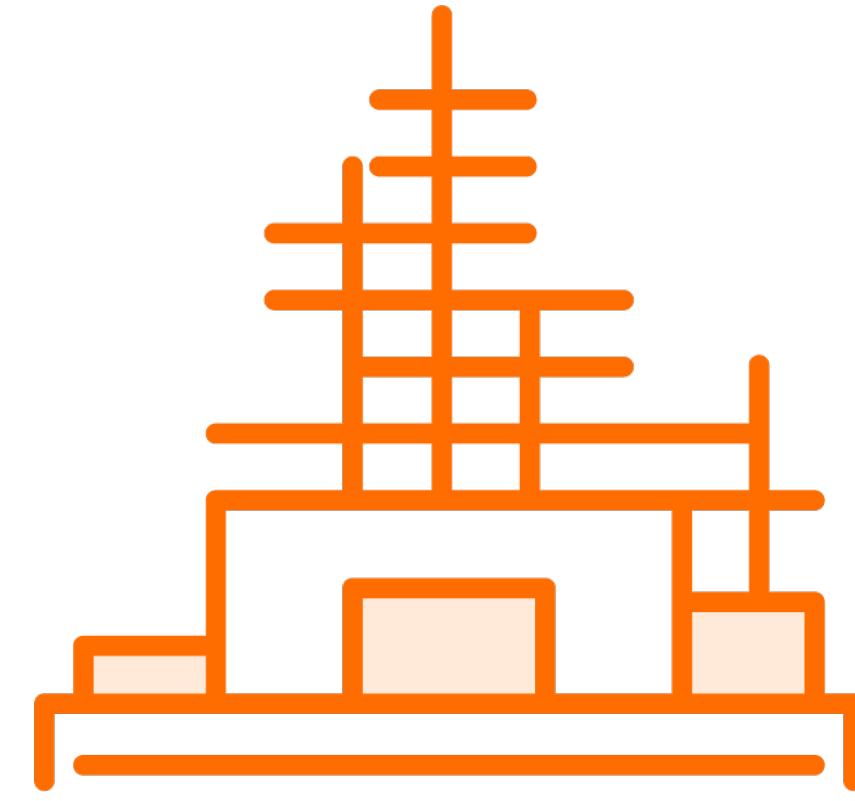
Act

Exercise the unit under test

Assert

Make claims about what happened

In an Approval Test



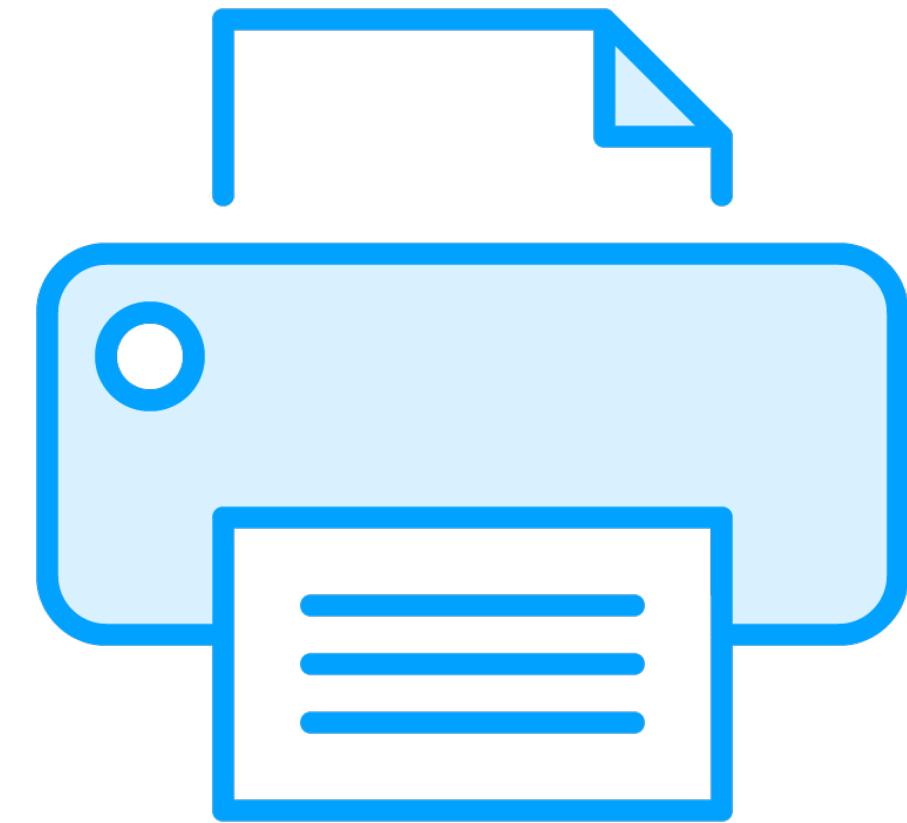
Arrange
Set up the object to
be tested, and
collaborators



Act
Exercise the unit
under test



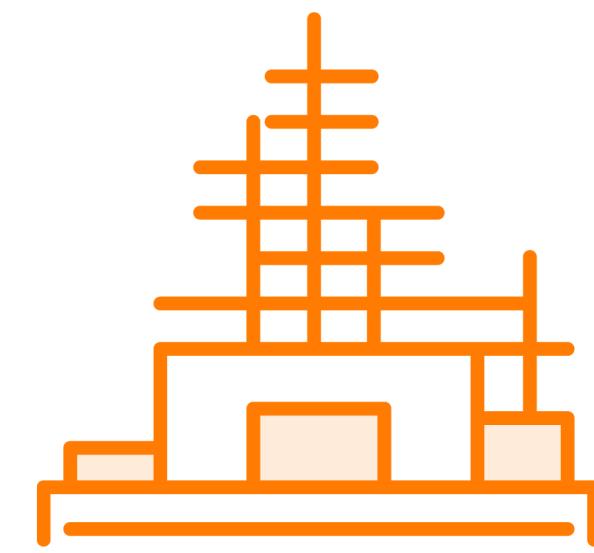
Diff
Check if what happened matches
the approved version



Displays results and domain objects as text

- Give a readable diff when a test fails
- The same printer can be used in several tests

Assert or Print Diff



Arrange



Act



Print



Assert

OR



Diff

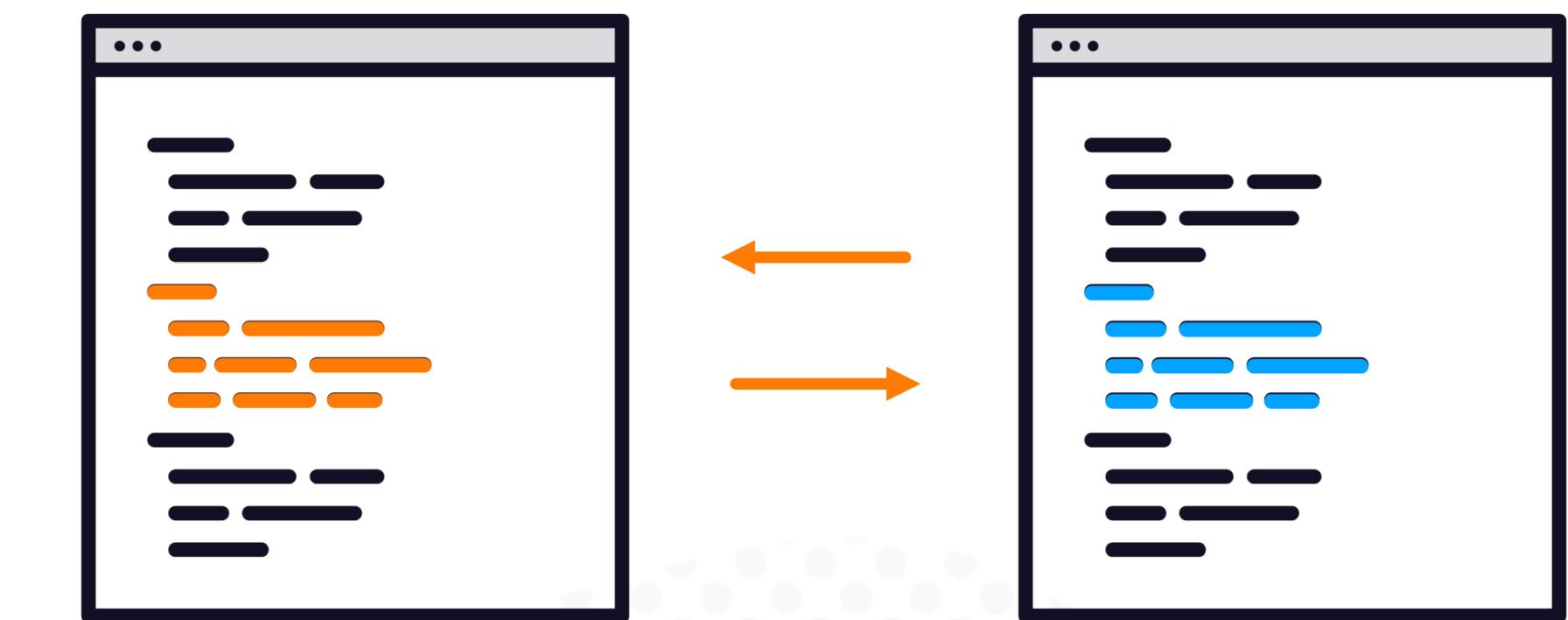


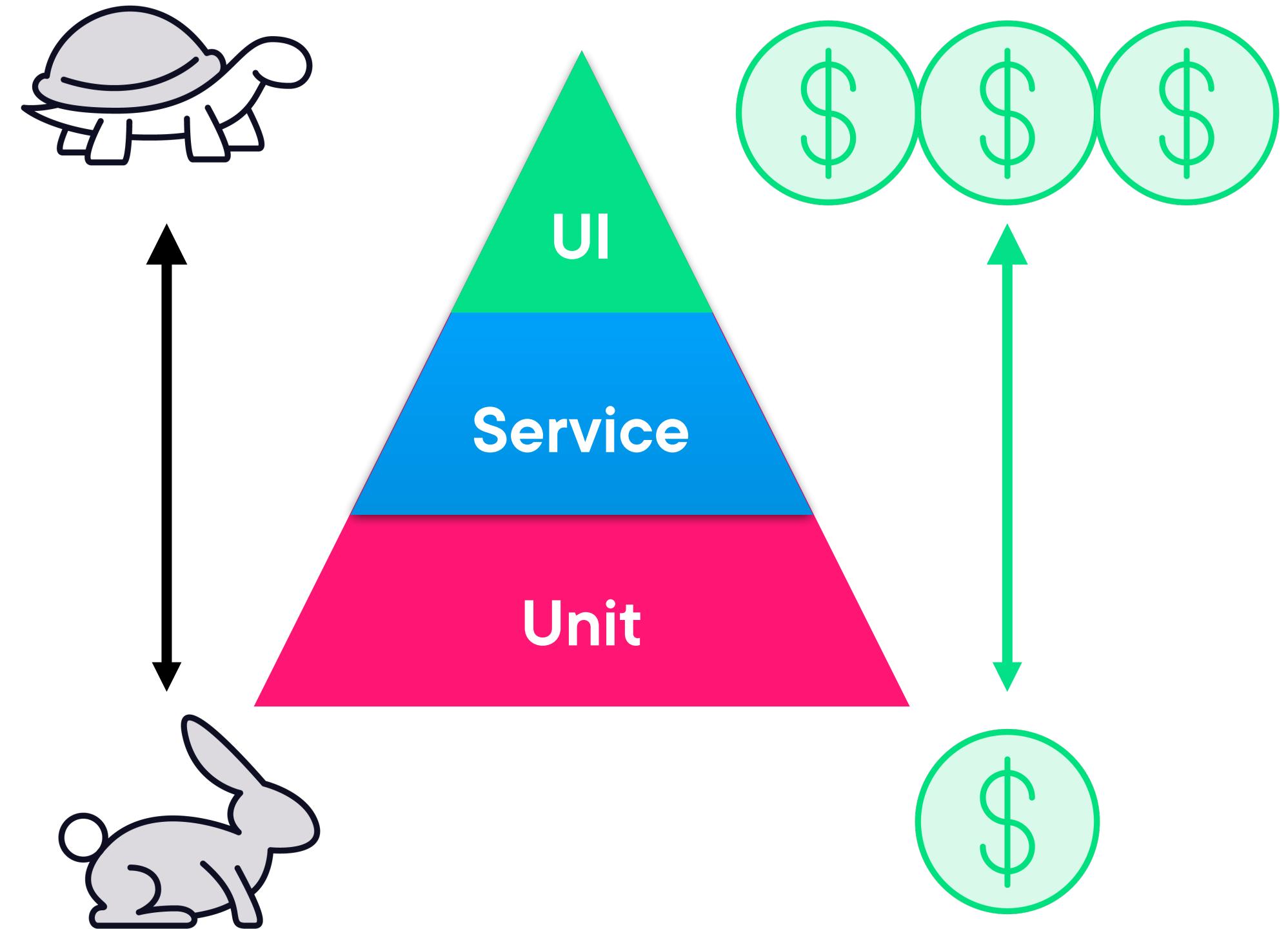
Test Failure: Inspect Using a Diff

Easier to understand failures with a diff

Easier to update approved files with a tool

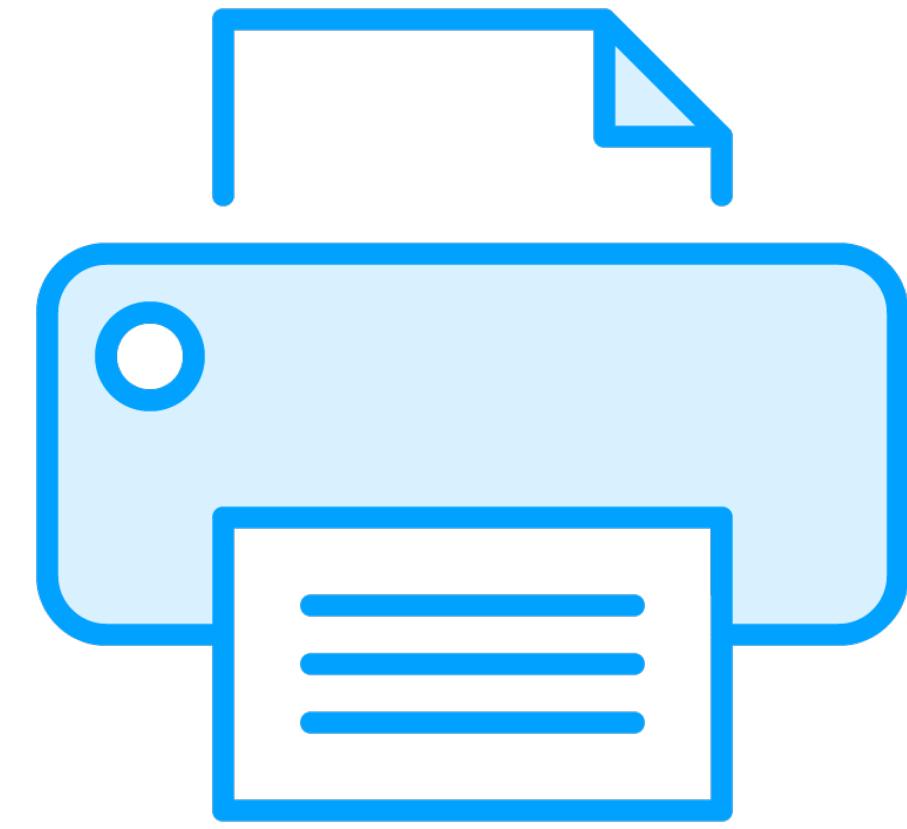
Cheaper test maintenance





Use Approval testing with a bigger chunk of code

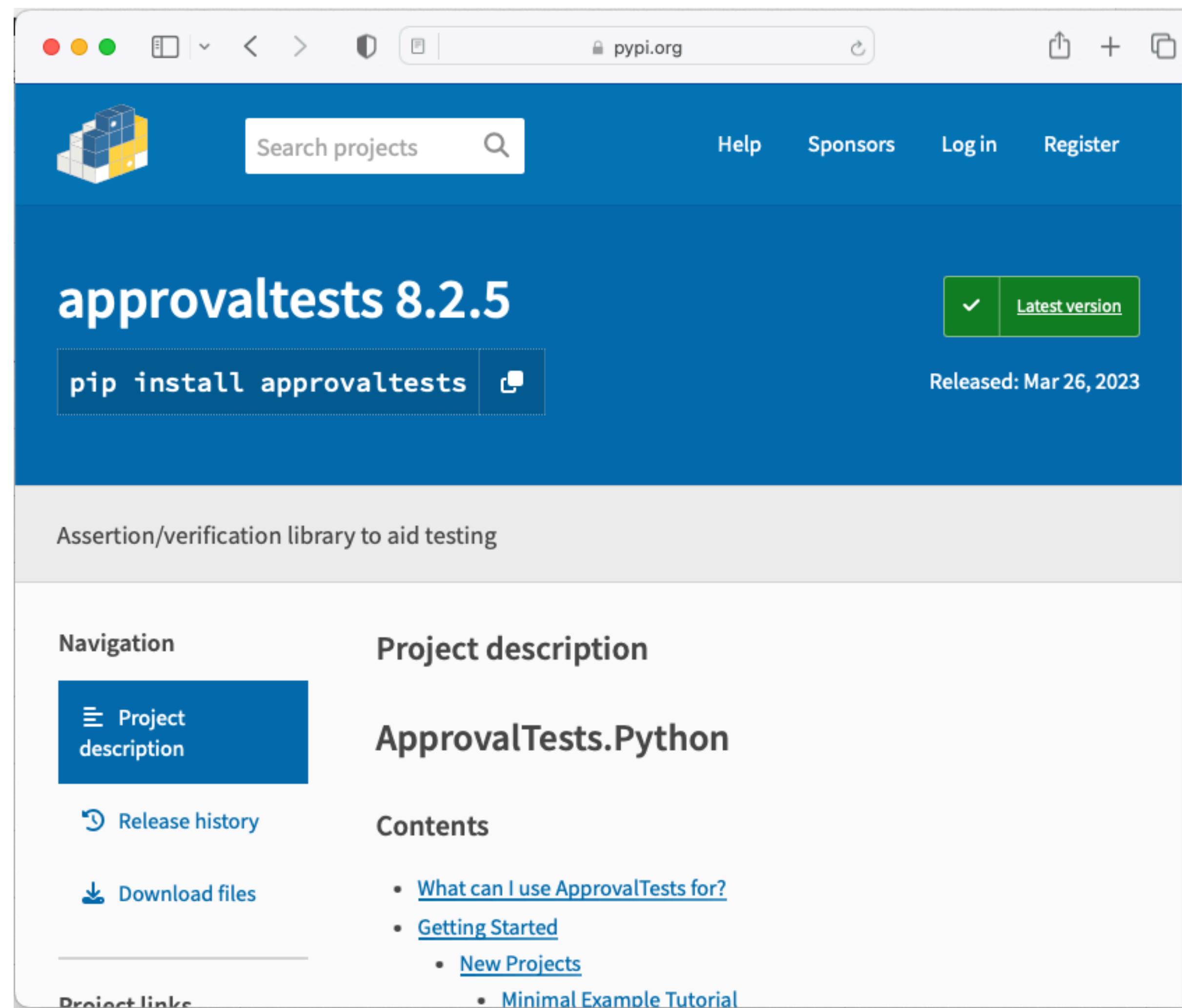
- When you need more than one assertion



Printer Design:

- Lay out text on multiple lines
- Exclude irrelevant details

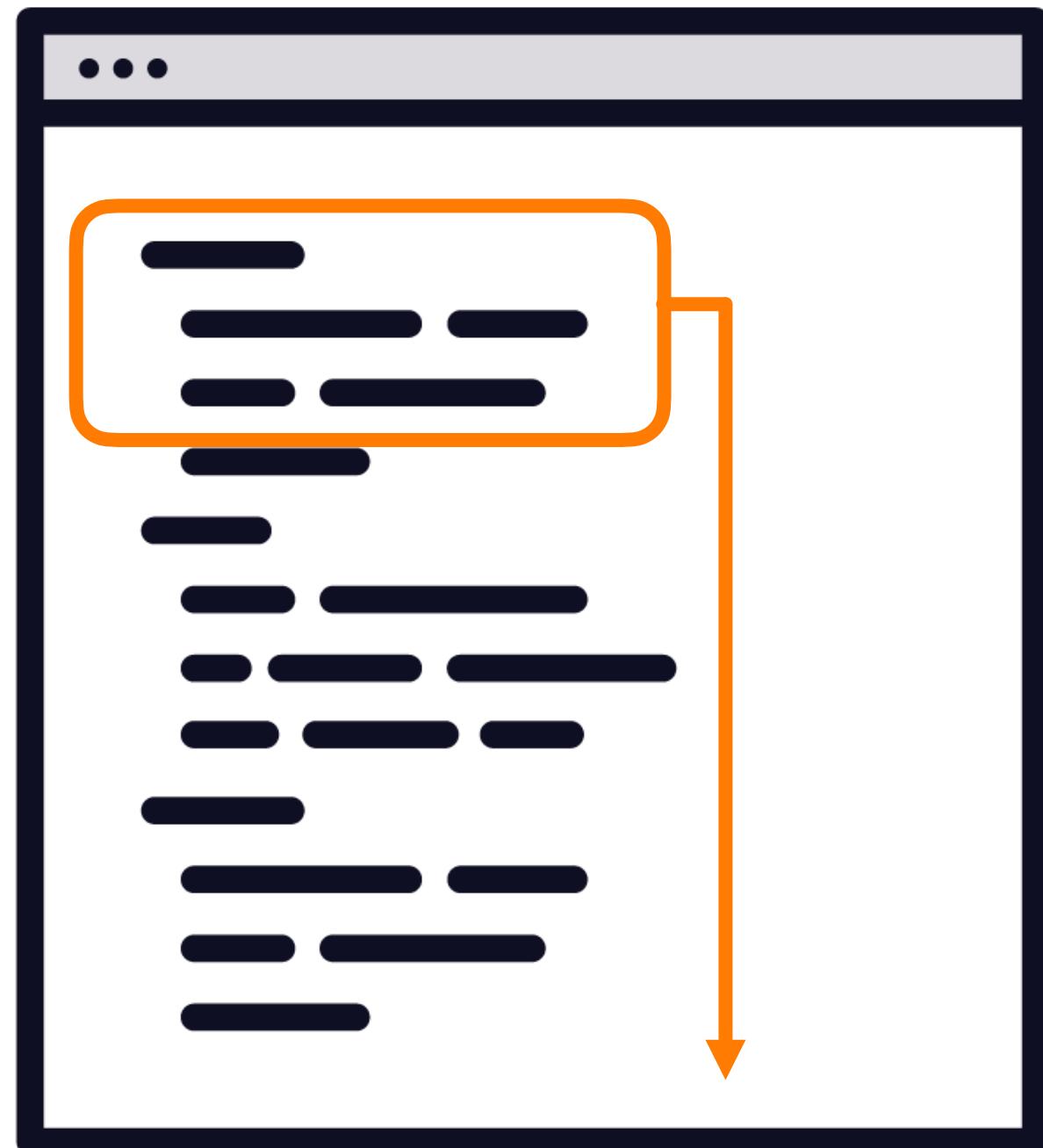
Use a Tool to update Approved files



Demo - Supermarket Receipt

Code Coverage

Measuring Coverage



Shows which lines are executed by your tests
If a line is not executed, it could have a bug in

Using Test Coverage

```
test_supermarket.py × shopping_cart.py ×
29
30     def handle_offers(self, receipt, offers, catalog):
31         for p in self._product_quantities.keys():
32             quantity = self._product_quantities[p]
33             if p in offers.keys():
34                 offer = offers[p]
35                 unit_price = catalog.unit_price(p)
36                 quantity_as_int = int(quantity)
37                 discount = None
38                 x = 1
39                 if offer.offer_type == SpecialOfferType.THREE_FOR_TWO:
40                     x = 3
41
42                 elif offer.offer_type == SpecialOfferType.TWO_FOR_AMOUNT:
43                     x = 2
44                     if quantity_as_int >= 2:
45                         total = offer.argument * (quantity_as_int / x) + quantity_as_int % 2 * unit_price
46                         discount_n = unit_price * quantity - total
47                         discount = Discount(p, "2 for " + str(offer.argument), -discount_n)
48
49                     if offer.offer_type == SpecialOfferType.FIVE_FOR_AMOUNT:
50                         x = 5
51
52                     number_of_x = math.floor(quantity_as_int / x)
53                     if offer.offer_type == SpecialOfferType.THREE_FOR_TWO and quantity_as_int > 2:
```

Demo - Gilded Rose

Combination Approvals

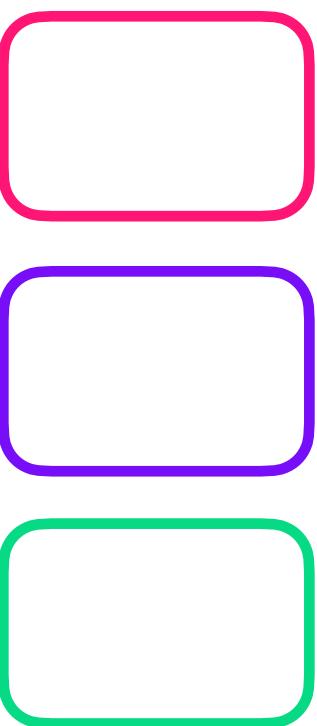
Combination Approvals

Test All Combinations of the Parameters
Print the output

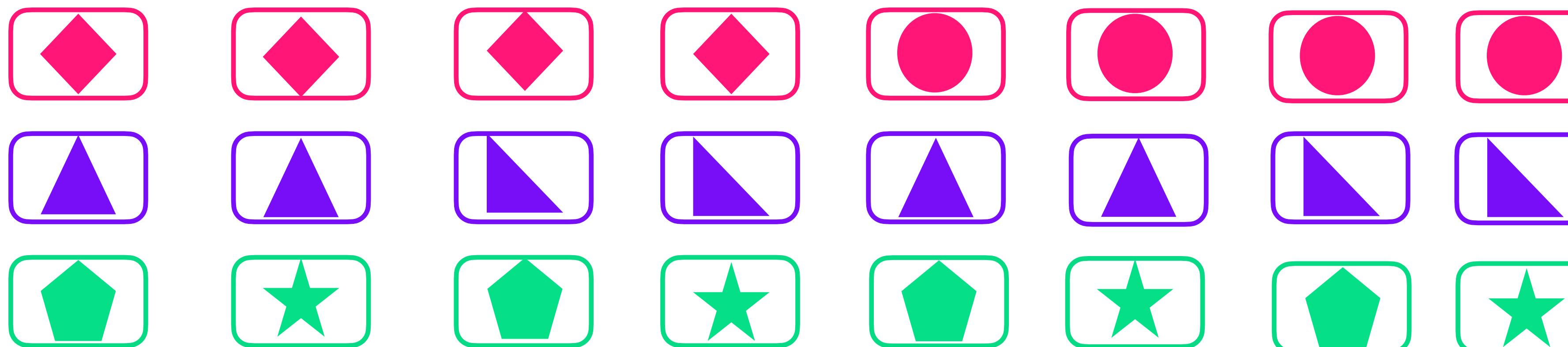
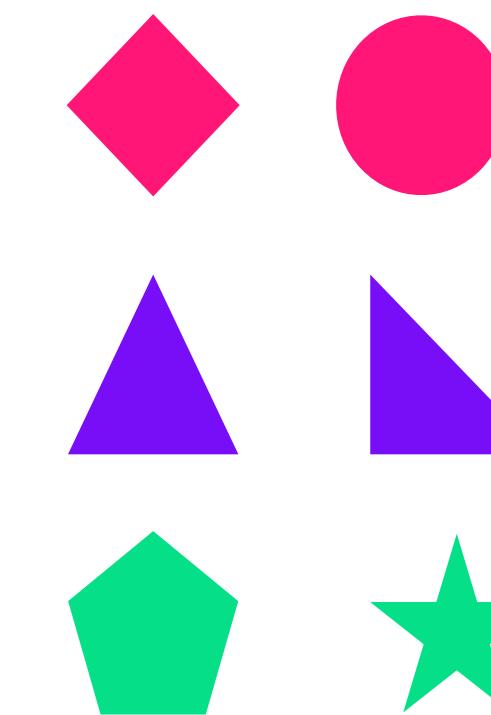
All Combinations

**With 3 parameters,
2 variants for each
→ 8 combinations**

Parameters



Available Values

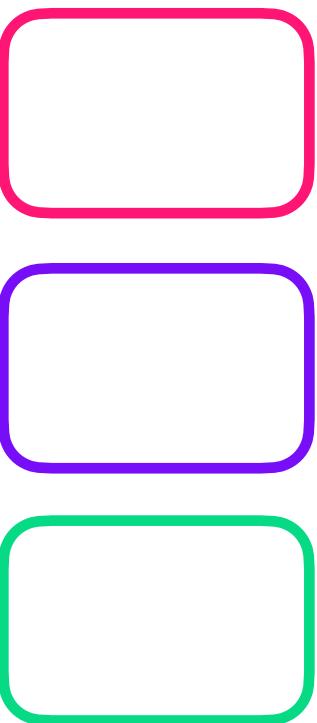


All Combinations

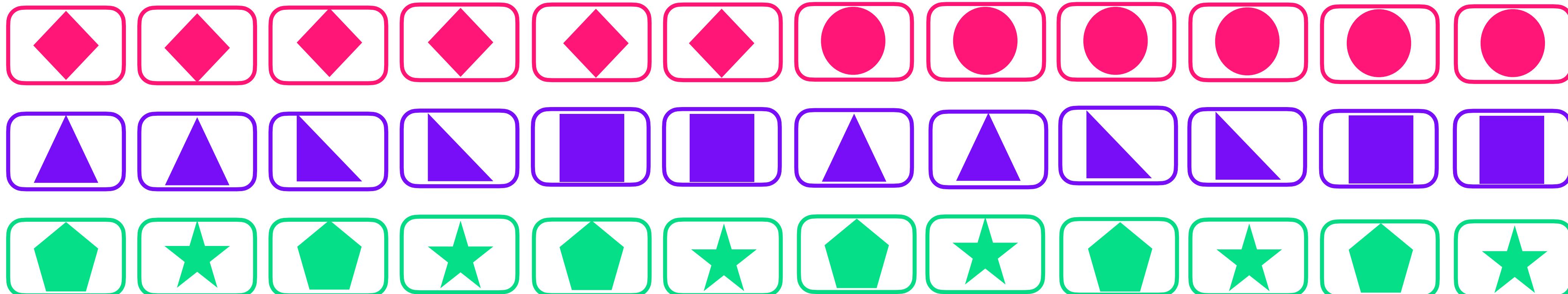
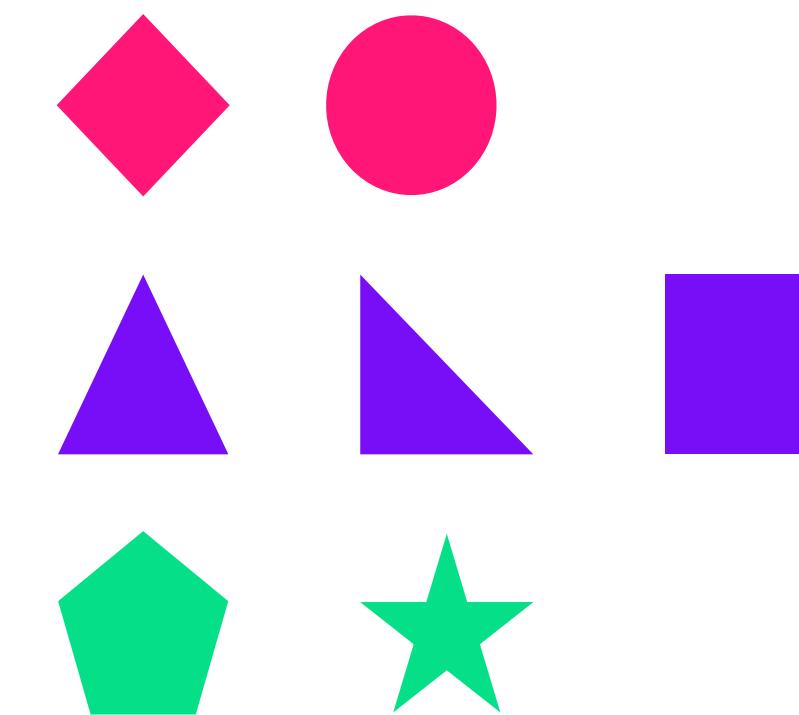
All Combinations

**With 3 parameters,
2,3,2 variants
→ 12 combinations**

Parameters



Available Values

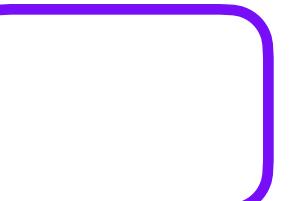
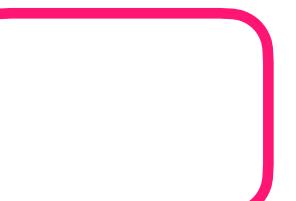


All Combinations

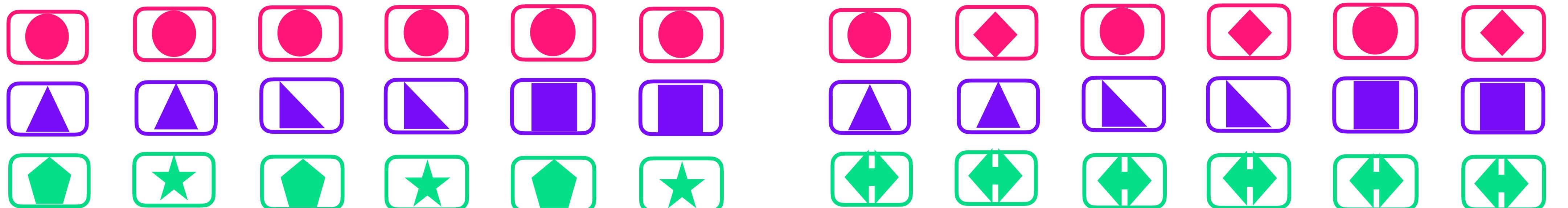
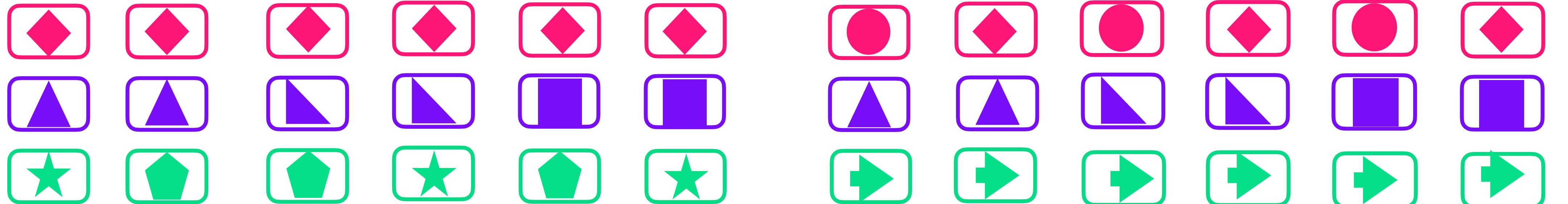
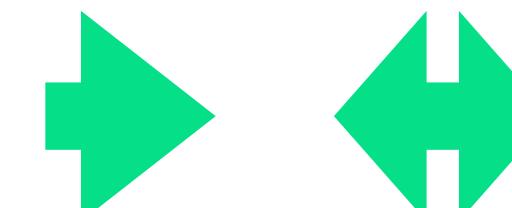
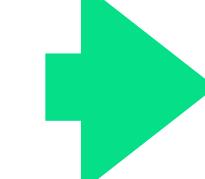
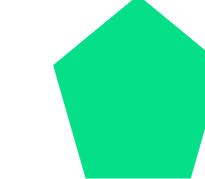
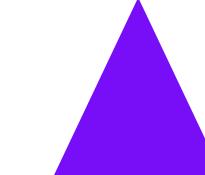
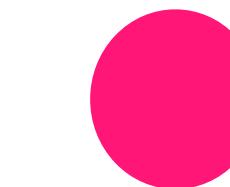
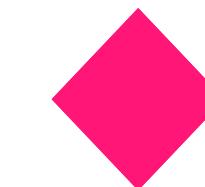
All Combinations

**With 3 parameters,
2,3,4 variants
→ 24 combinations**

Parameters



Available Values



Demo - Gilded Rose

**Mutation Testing
Branch Coverage**

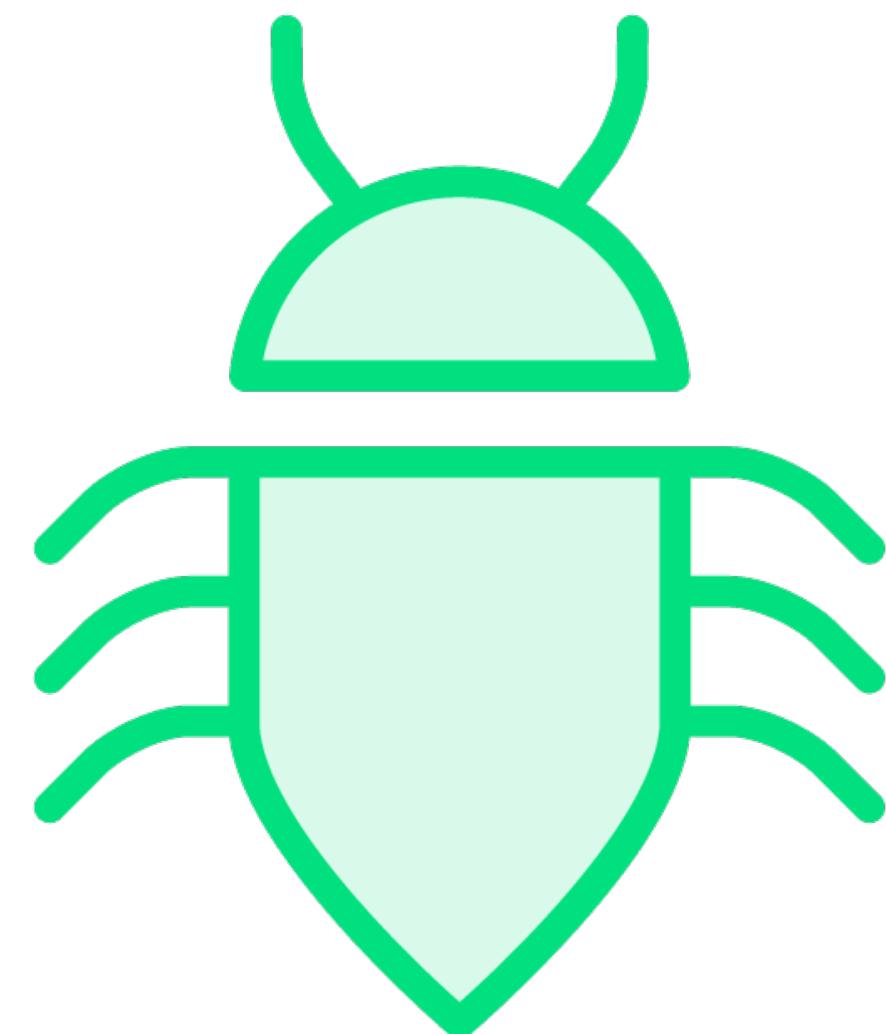
Mutation Testing

Deliberately
insert a bug

Run the tests and
see if they fail

If the tests DON'T
fail, then there is
a problem

Mutation Testing - Drawbacks



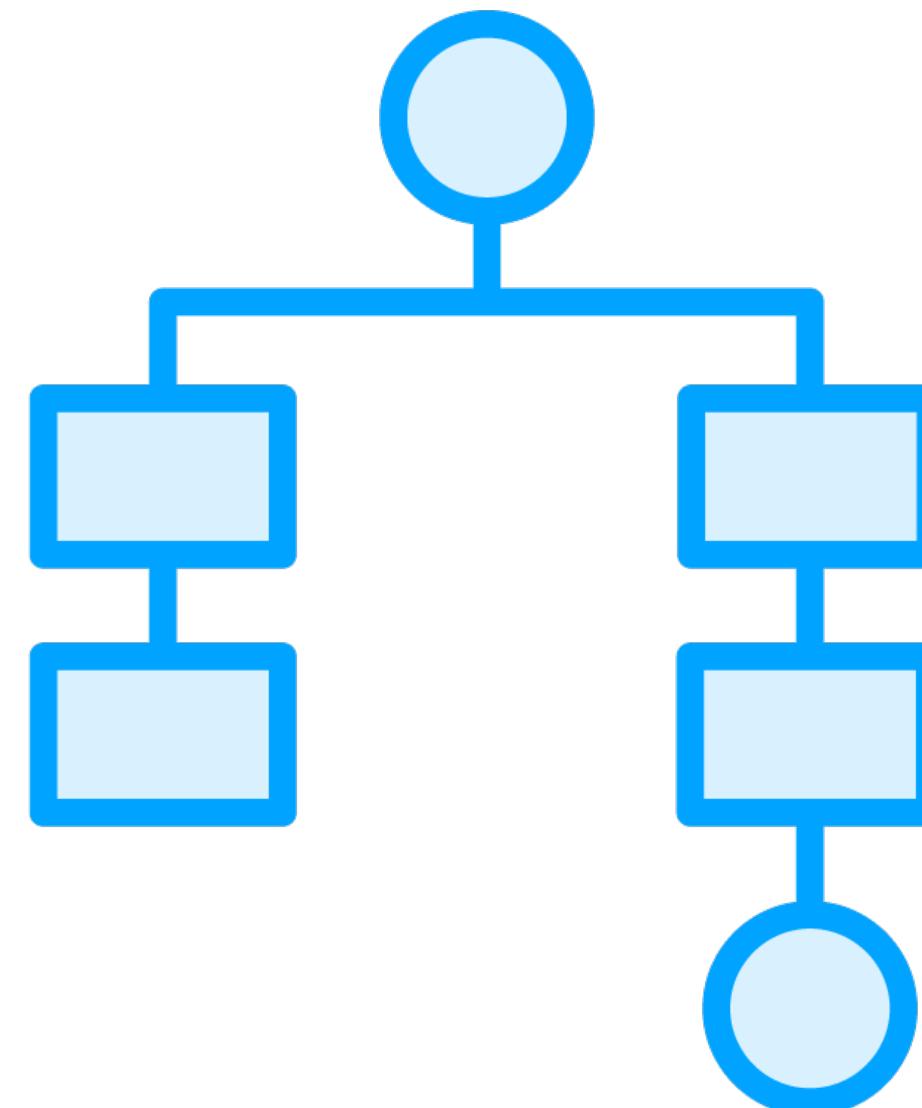
Tools only know certain kinds of bugs to insert
Time and resource heavy

Branch Coverage

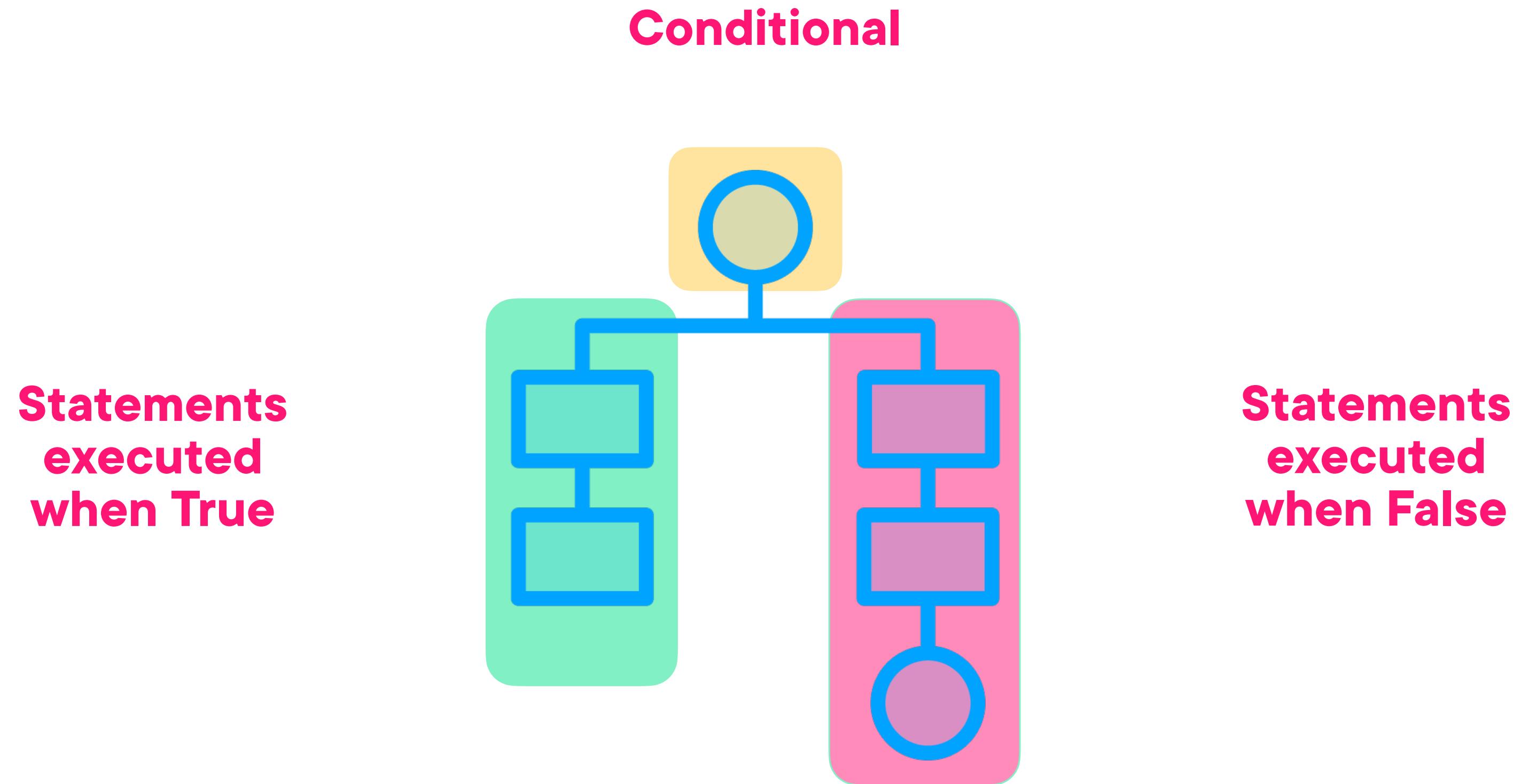
Conditional

Statements
executed
when True

Statements
executed
when False

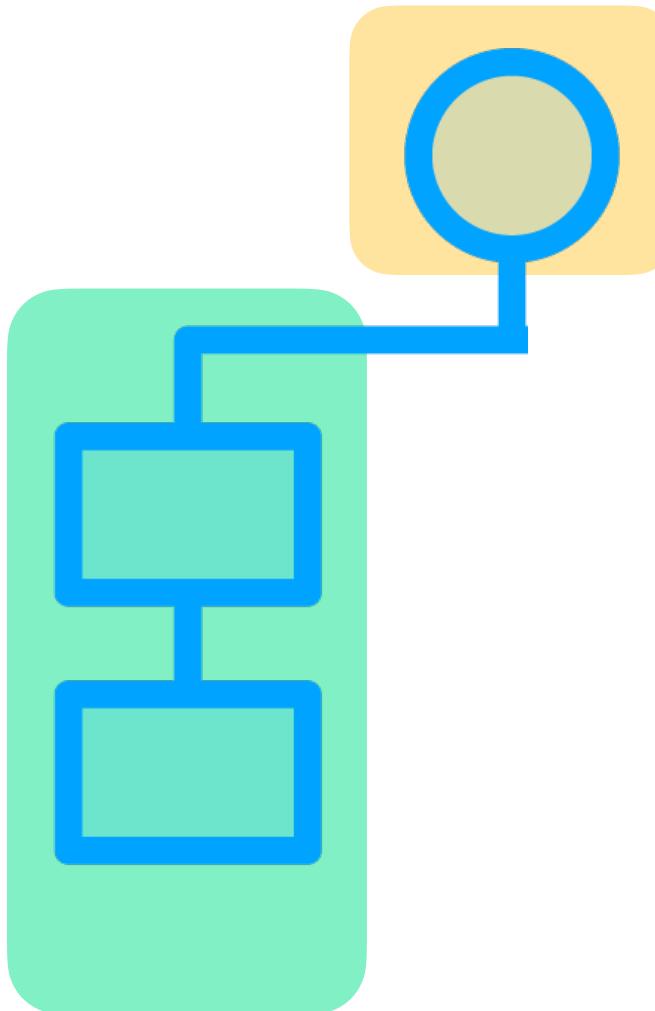


Branch Coverage



Branch Coverage

Statements
executed
when True



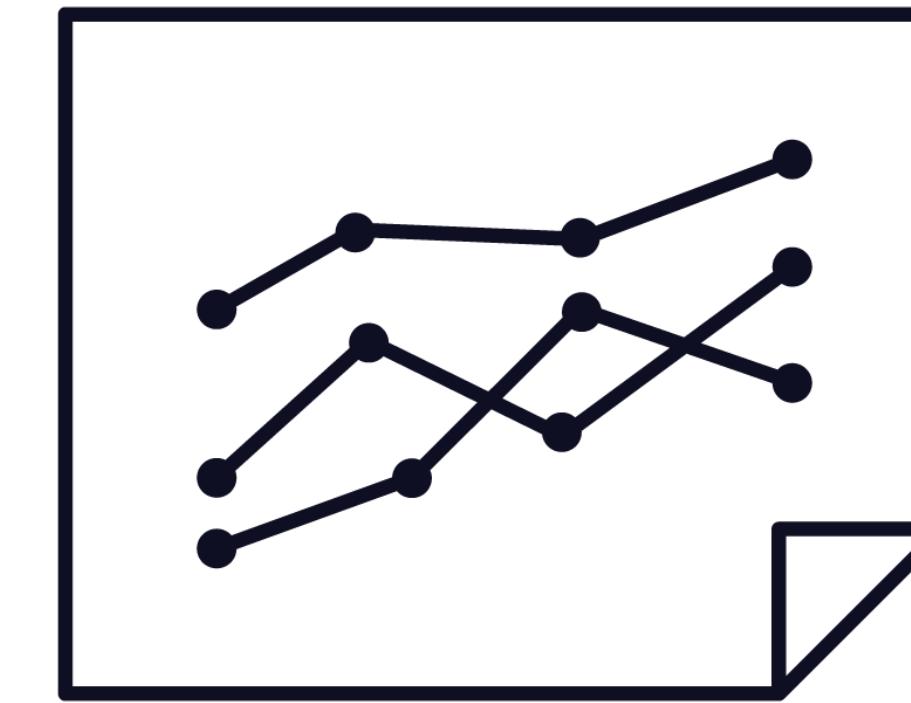
Conditional

No alternative
statement

Situations to Use Coverage & Mutation testing

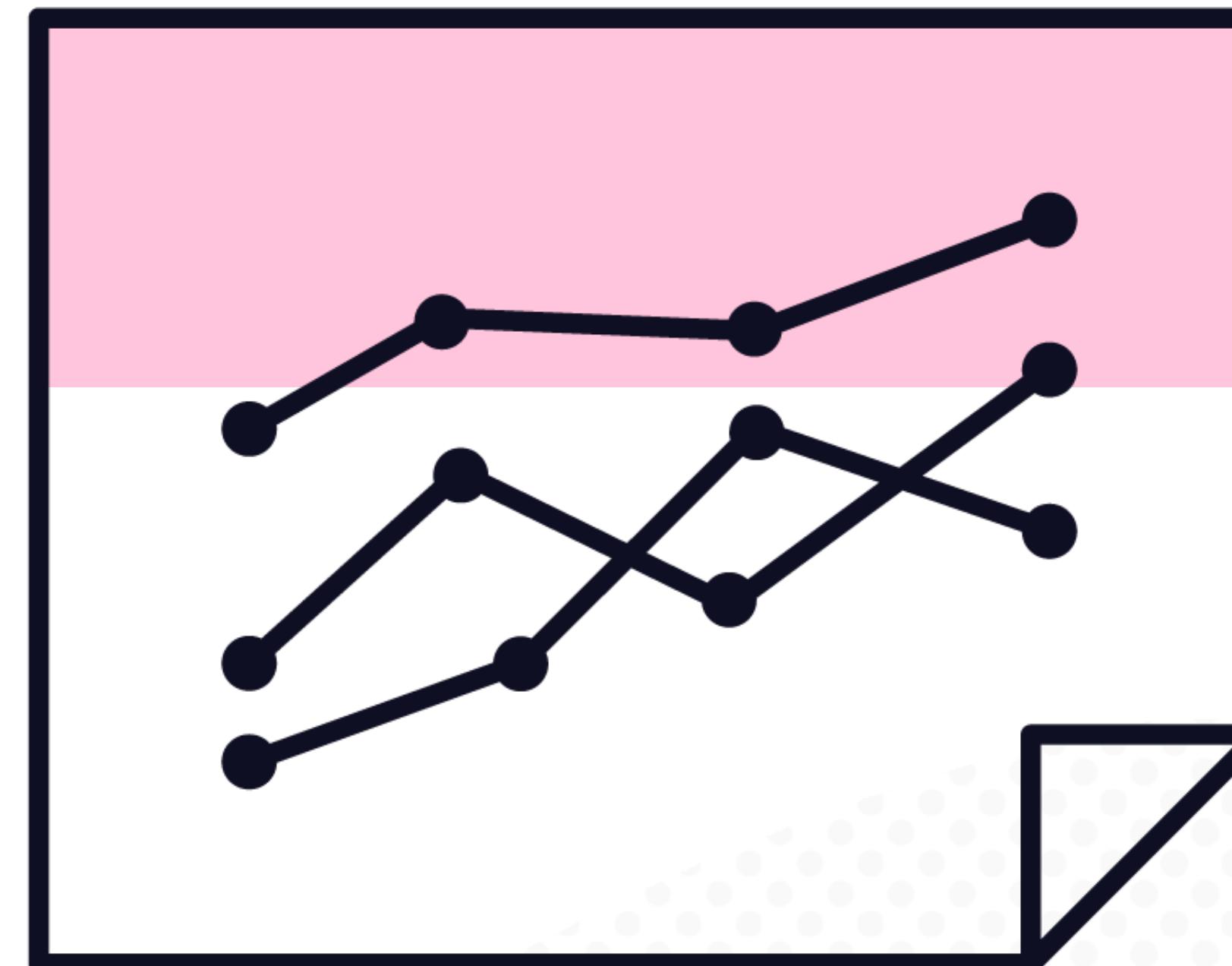


Spot missing tests



Track trends
over time

Coverage Targets



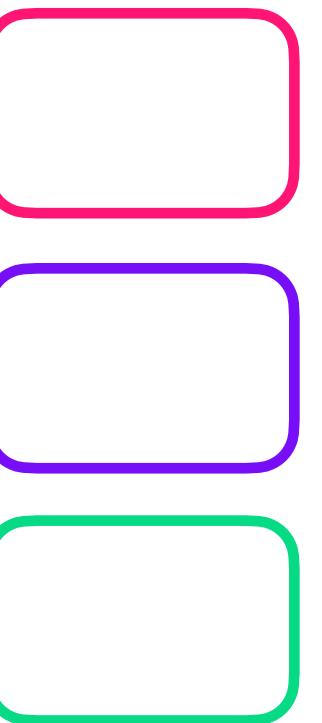
Demo - Gilded Rose

Pairwise Testing

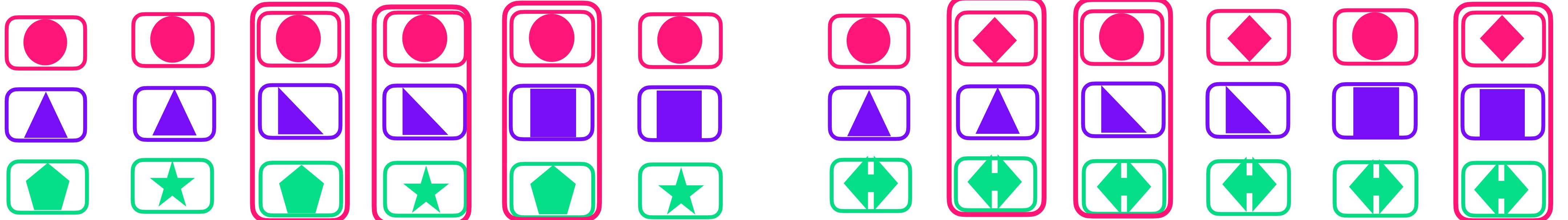
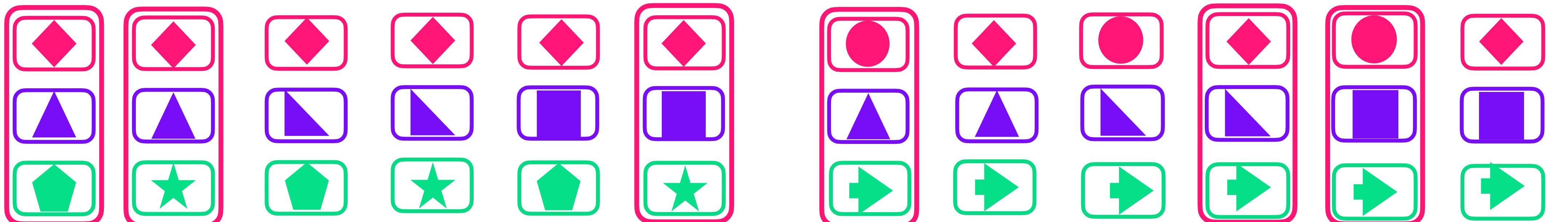
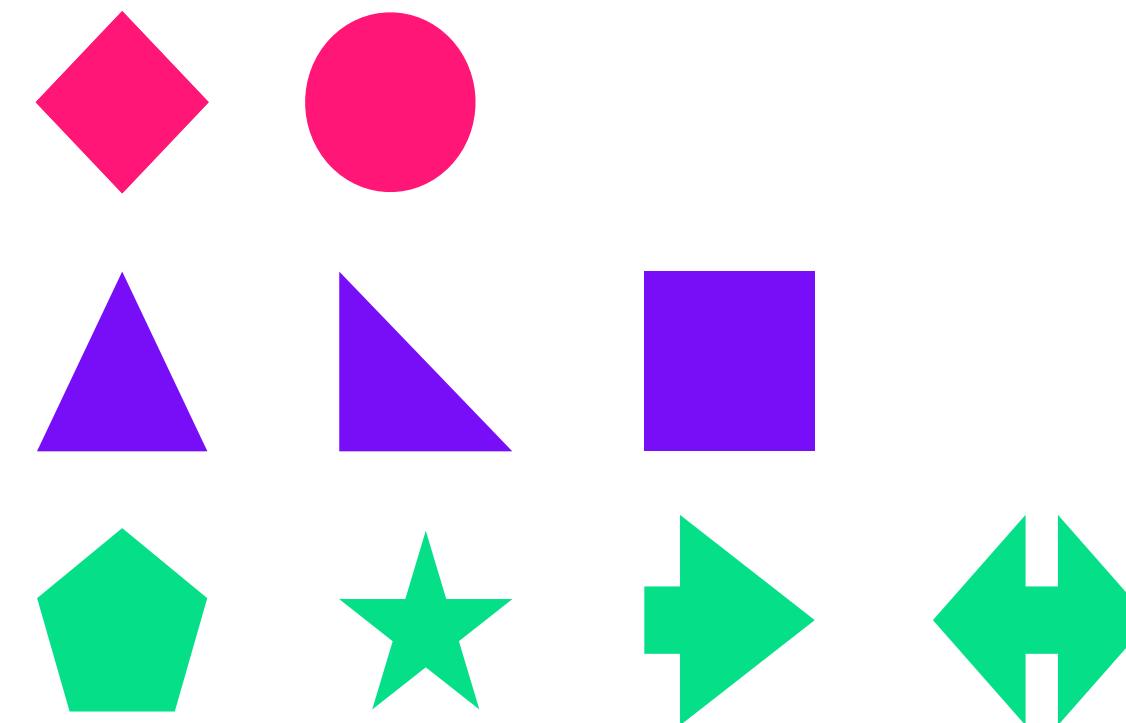
All Pairs

With 3 parameters,
2,3,4 variants
→ 12 unique pairs

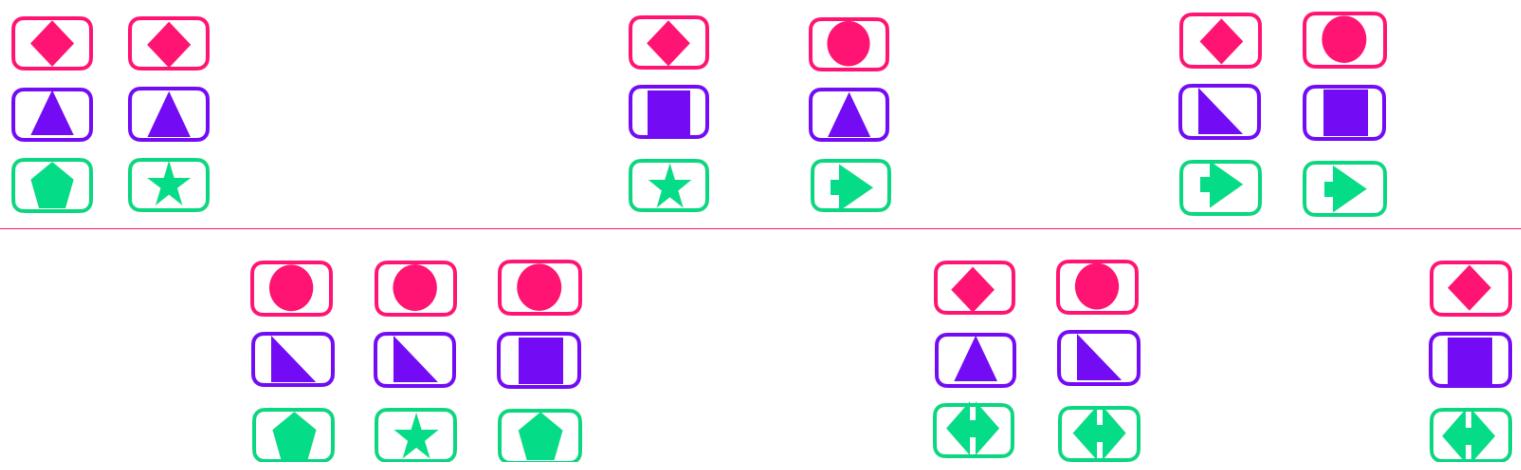
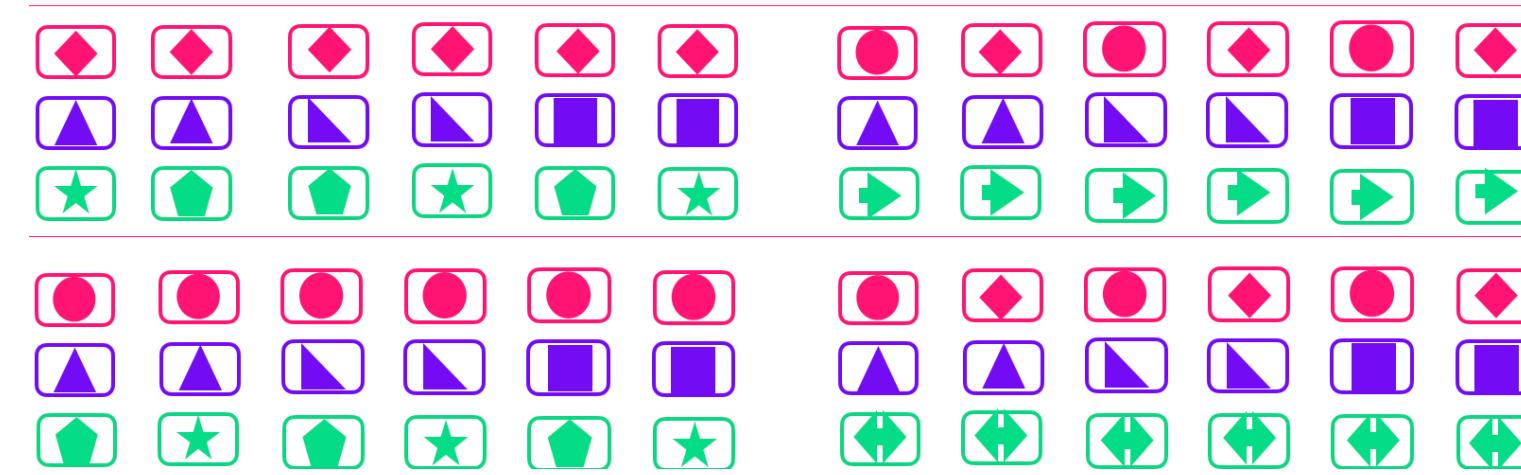
Parameters



Available Values



All Combinations, All Pairs



Combination testing improves coverage

Approval Testing makes it easier to manage

All-Pairs reduces the number of combinations

Summary

Approval Testing

Code Coverage

All Combinations

Mutation Testing