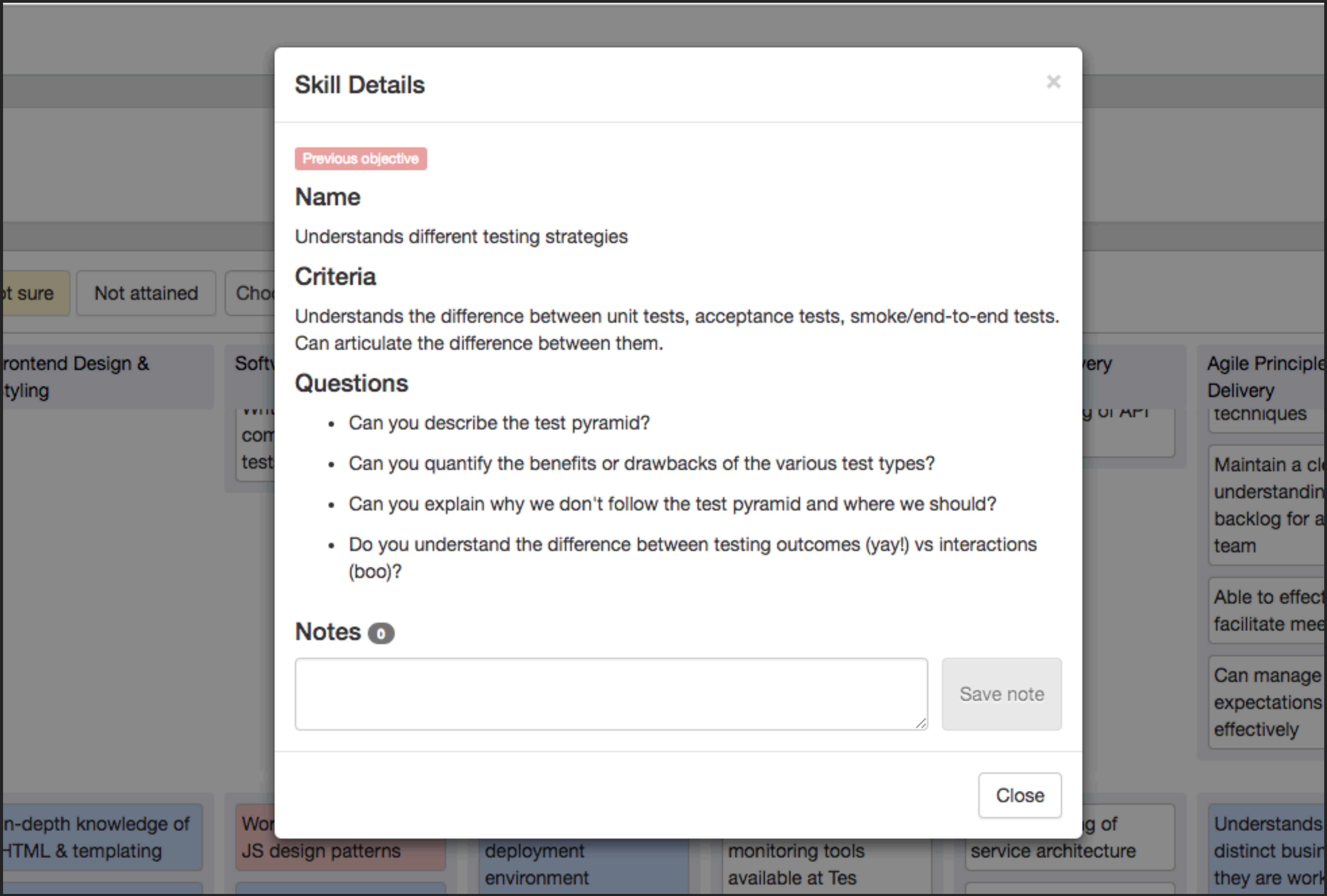
A SKILLS-MAP EXPLANATION OF…

THE TESTING PYRAMID

WE NEED TO KNOW…



×
Previous objective
Save note
Close
t sure
Not attained
Cho
rontend Design & tyling
Soft
Agile Principle Delivery
techniques
Maintain a cl understandin backlog for a team
Able to effect facilitate mee
Can manage expectations effectively
n-depth knowledge of HTML & templating
Wor JS design patterns
deployment environment
monitoring tools available at Tes
g of service architecture
Understands distinct busin they are work

Skill Details

Previous objective

Name

Understands different testing strategies

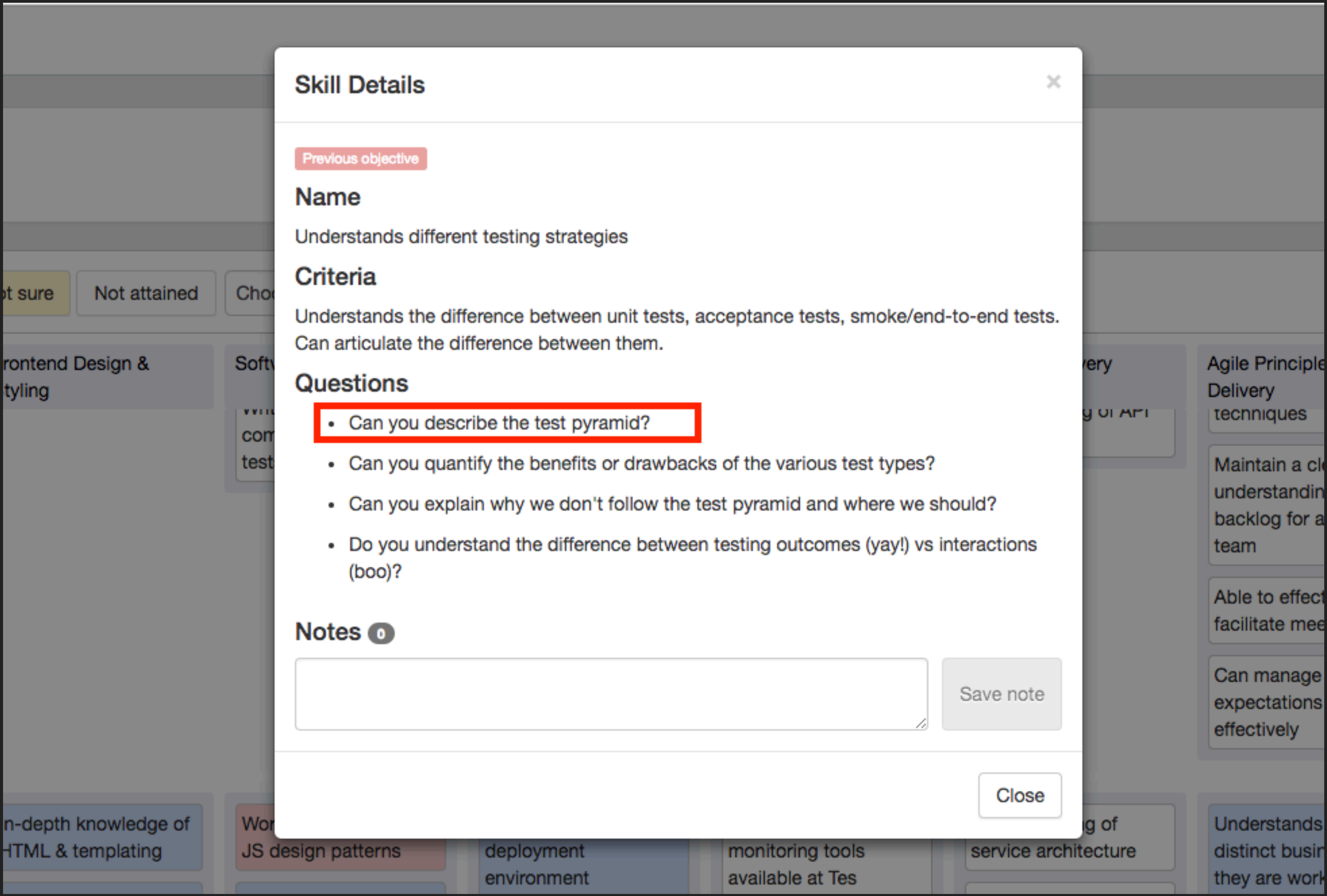
Criteria

Understands the difference between unit tests, acceptance tests, smoke/end-to-end tests. Can articulate the difference between them.

Questions

- Can you describe the test pyramid?
- Can you quantify the benefits or drawbacks of the various test types?
- Can you explain why we don't follow the test pyramid and where we should?
- Do you understand the difference between testing outcomes (yay!) vs interactions (boo)?

Notes 0



Skill Details

Name

Understands different testing strategies

Criteria

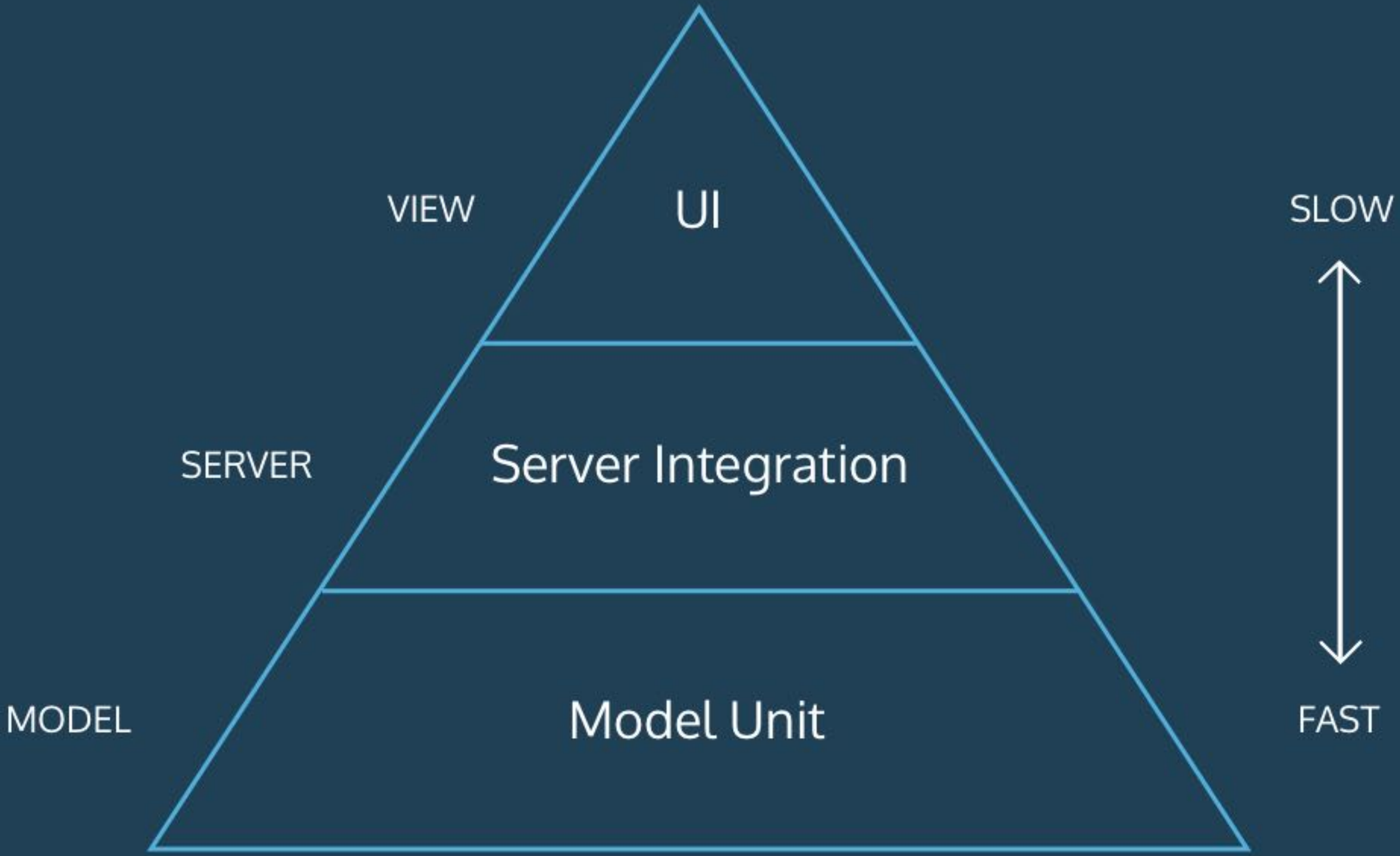
Understands the difference between unit tests, acceptance tests, smoke/end-to-end tests. Can articulate the difference between them.

Questions

- Can you describe the test pyramid?
- Can you quantify the benefits or drawbacks of the various test types?
- Can you explain why we don't follow the test pyramid and where we should?
- Do you understand the difference between testing outcomes (yay!) vs interactions (boo)?

Notes 0

Save note
Close
t sure
Not attained
Cho
rontend Design & tyling
Soft
Agile Principle Delivery
techniques
Maintain a cl understandin backlog for a team
Able to effect facilitate mee
Can manage expectations effectively
n-depth knowledge of HTML & templating
Wor JS design patterns
deployment environment
monitoring tools available at Tes
service architecture
Understands distinct busi they are work



VIEW

UI

SERVER

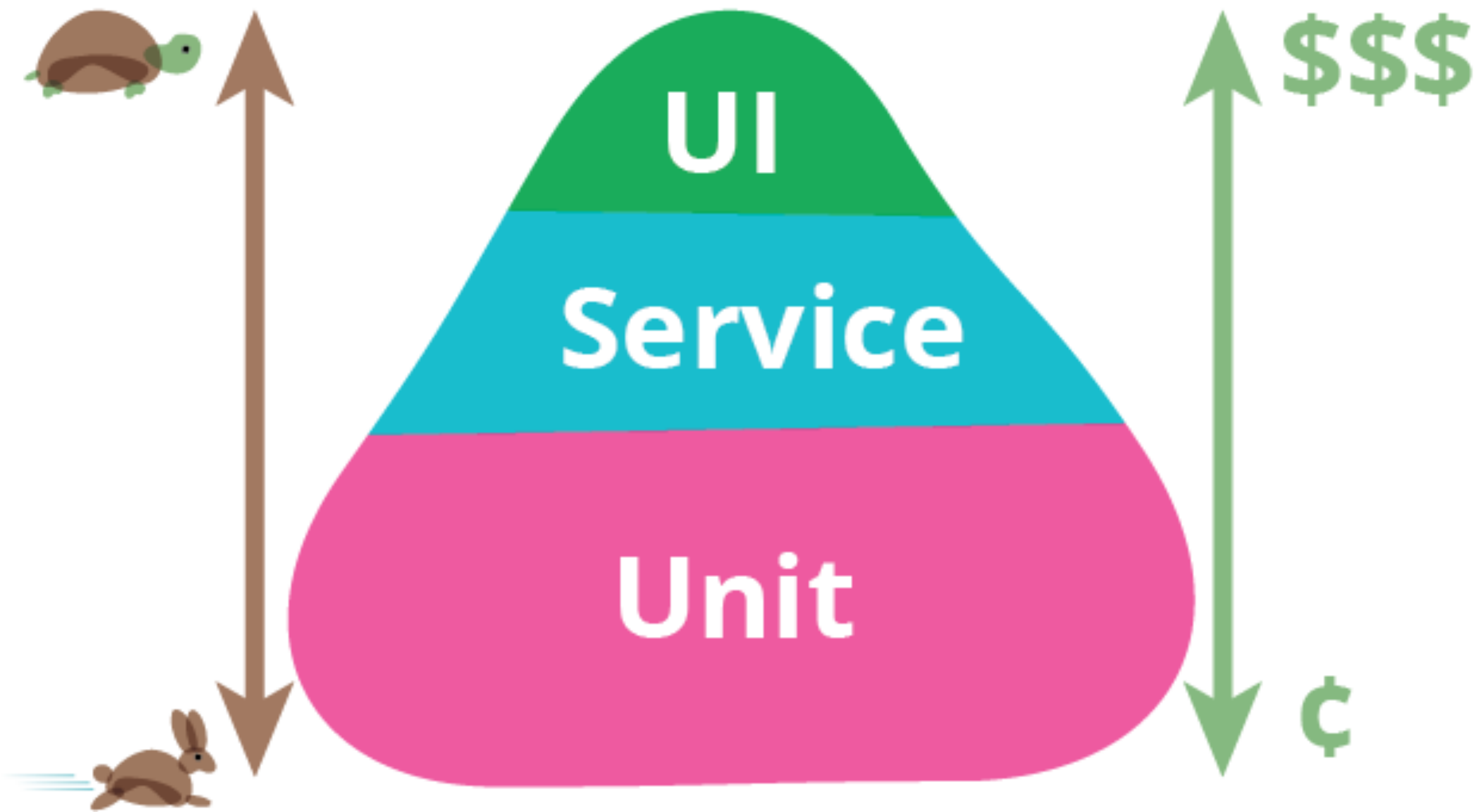
Server Integration

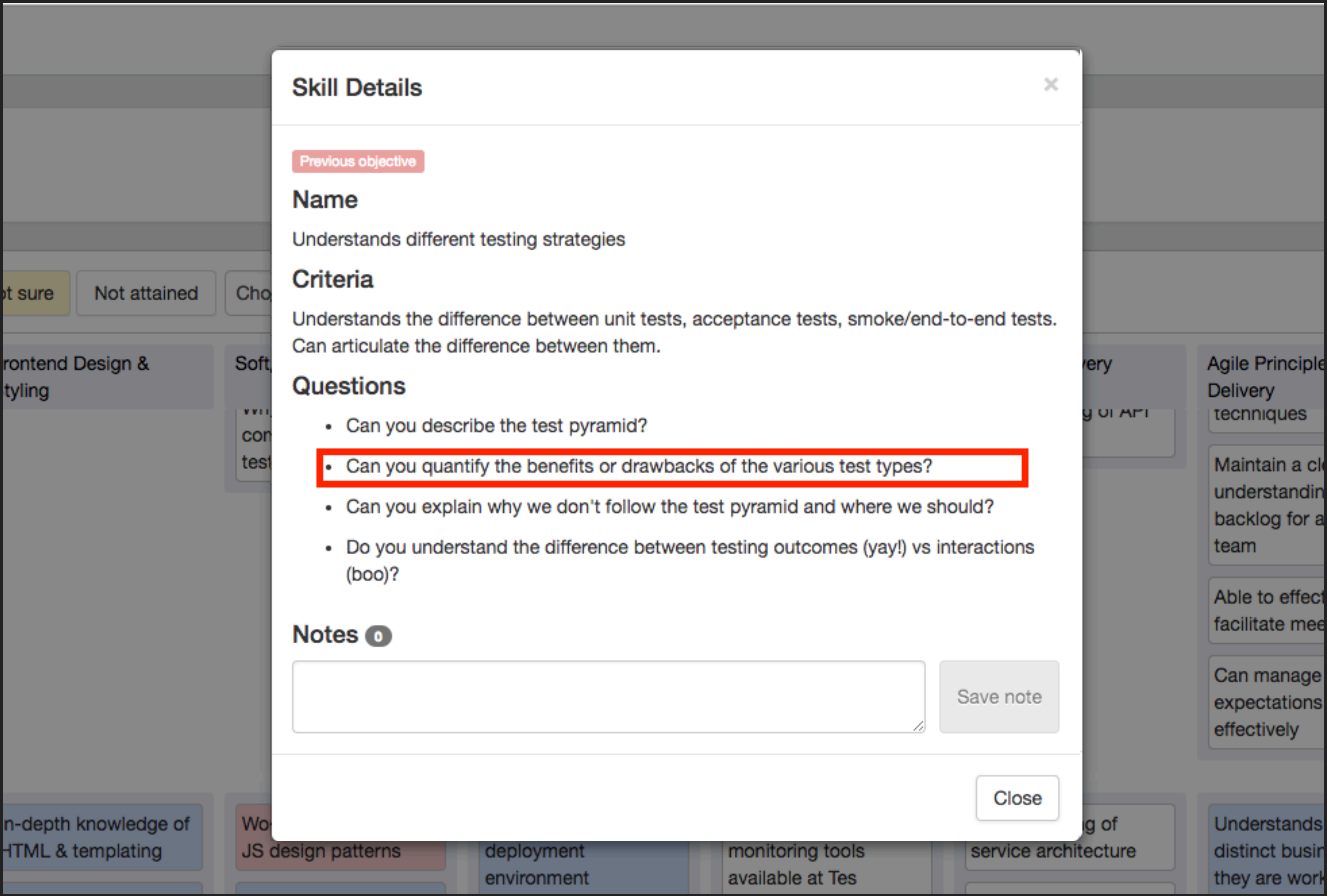
MODEL

Model Unit

SLOW

FAST


$$$
UI
Service
Unit
¢



Skill Details

×

Previous objective

Name

Understands different testing strategies

Criteria

Understands the difference between unit tests, acceptance tests, smoke/end-to-end tests. Can articulate the difference between them.

Questions

- Can you describe the test pyramid?
- Can you quantify the benefits or drawbacks of the various test types?
- Can you explain why we don't follow the test pyramid and where we should?
- Do you understand the difference between testing outcomes (yay!) vs interactions (boo)?

Notes 0

Save note

Close

UNIT TESTS

STRENGTHS

UNIT TESTS

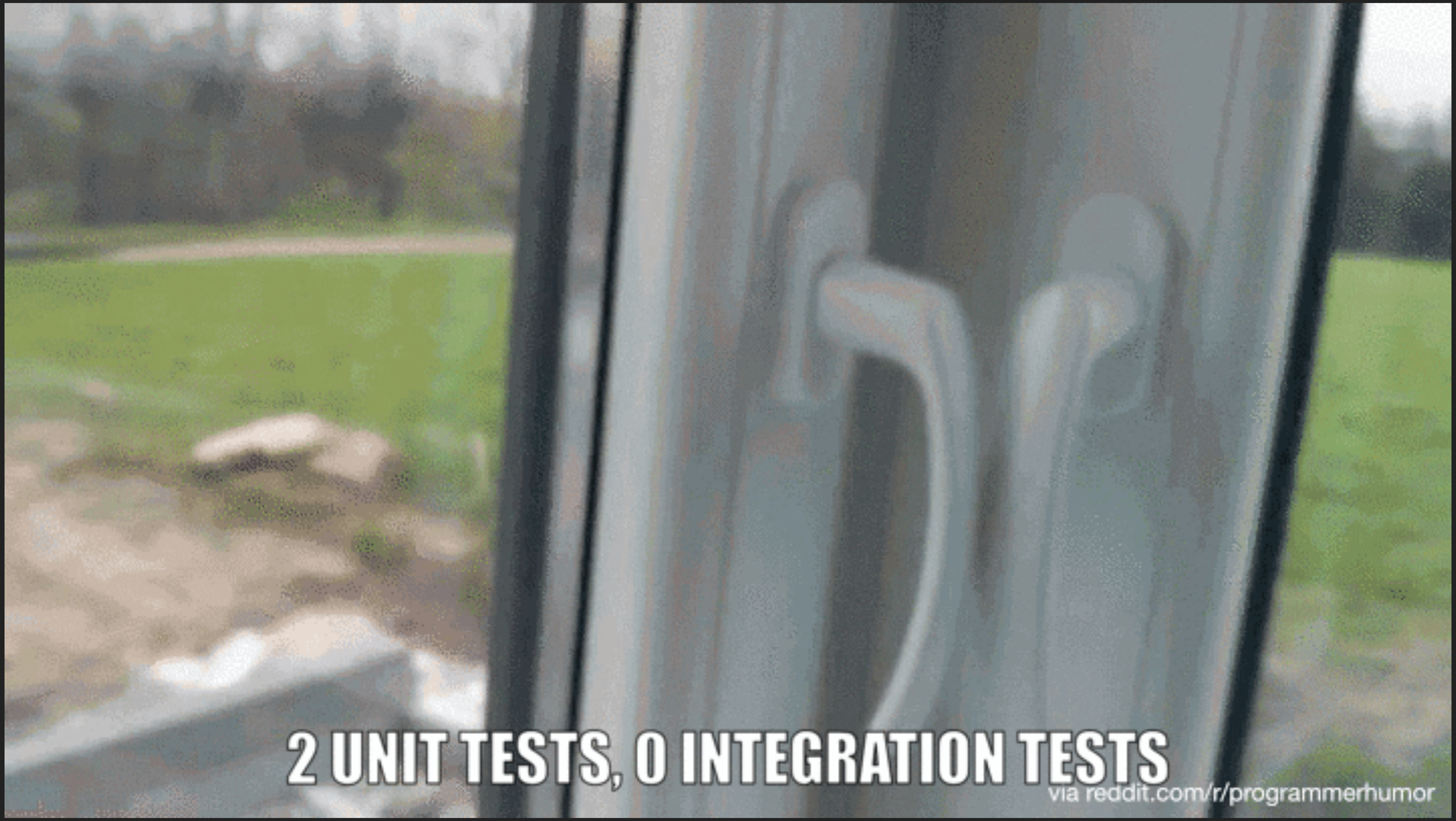
- Quick and cheap to write
- Find bugs easily
- Encourages and helps refactoring (Agile).
- Makes more complex integration easier later on
- Provides documentation

DRAWBACKS / LIMITATIONS

UNIT TESTS

- Can’t account for every single execution path.
- Time consuming.
- Can’t catch bugs that emerge from combining units.

INTEGRATION TESTS



via reddit.com/r/programmerhumor

INTEGRATION TESTS

- Can be understood in lots of different ways.
- General principle that individual modules work as expected when combined together.
- Three common approaches:
    - Big Bang:
        - Integrate all module to create complete software system (high risk)
    - Bottom-up:
        - low level components -> high level components (efficient for error detection).
    - Top down:
        - High-level components -> low level components (detects lost module branch links).

INTEGRATION TESTS

- Big Bang:
  - Difficult to localise errors.
  - Can only be done after all modules have been designed.
  - Critical modules can’t be tested on priority.
- Bottom-up:
  - Critical components (at top level) are tested last, may be more prone to defects.
  - Early prototype not possible.
- Top down:
  - Requires lots of stubbing.
  - Lower level modules may be tested inadequately.

ACCEPTANCE TESTS

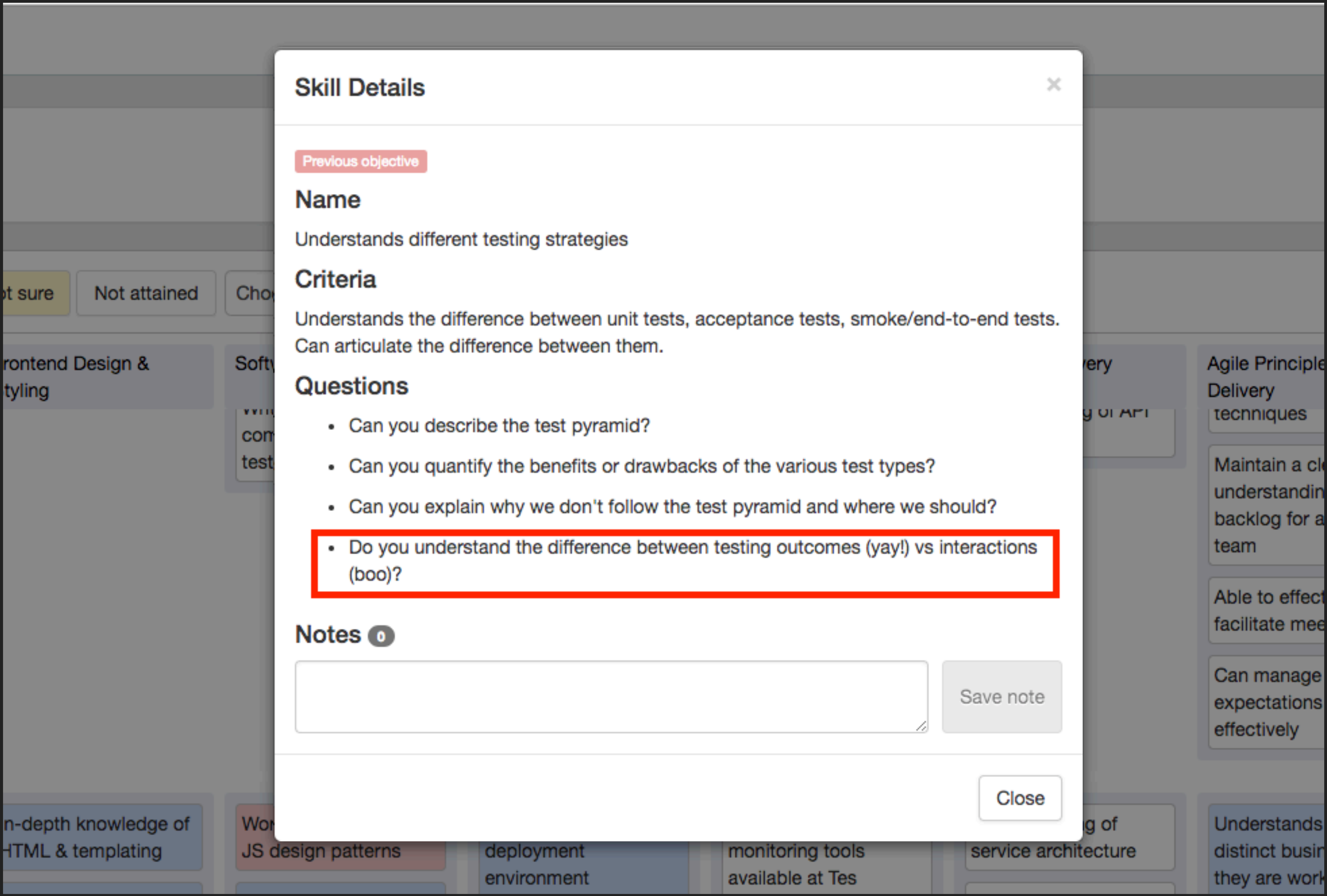
ACCEPTANCE TESTS

- Again, many different types.
- Can be thought of as “user story tests”.
- Checks that the entire service performs as expected.
- Usually driven by Black Box testing:
  - Tests have no knowledge of internal workings of systems.
- Ensures that our service is reliable and meets business needs.

ACCEPTANCE TESTS

- Very expensive in terms of time and resources.
- Quick to break.
- Not always clear if the benefits outweigh the costs.

OUTCOMES VS INTERACTIONS



×
Previous objective
Save note
Close
t sure
Not attained
Cho
rontend Design & tyling
Soft
Agile Principle Delivery
techniques
Maintain a cl understandin backlog for a team
Able to effect facilitate mee
Can manage expectations effectively
n-depth knowledge of HTML & templating
Wo JS design patterns
deployment environment
monitoring tools available at Tes
g of service architecture
Understands distinct busi they are work

Skill Details

Name

Understands different testing strategies

Criteria

Understands the difference between unit tests, acceptance tests, smoke/end-to-end tests. Can articulate the difference between them.

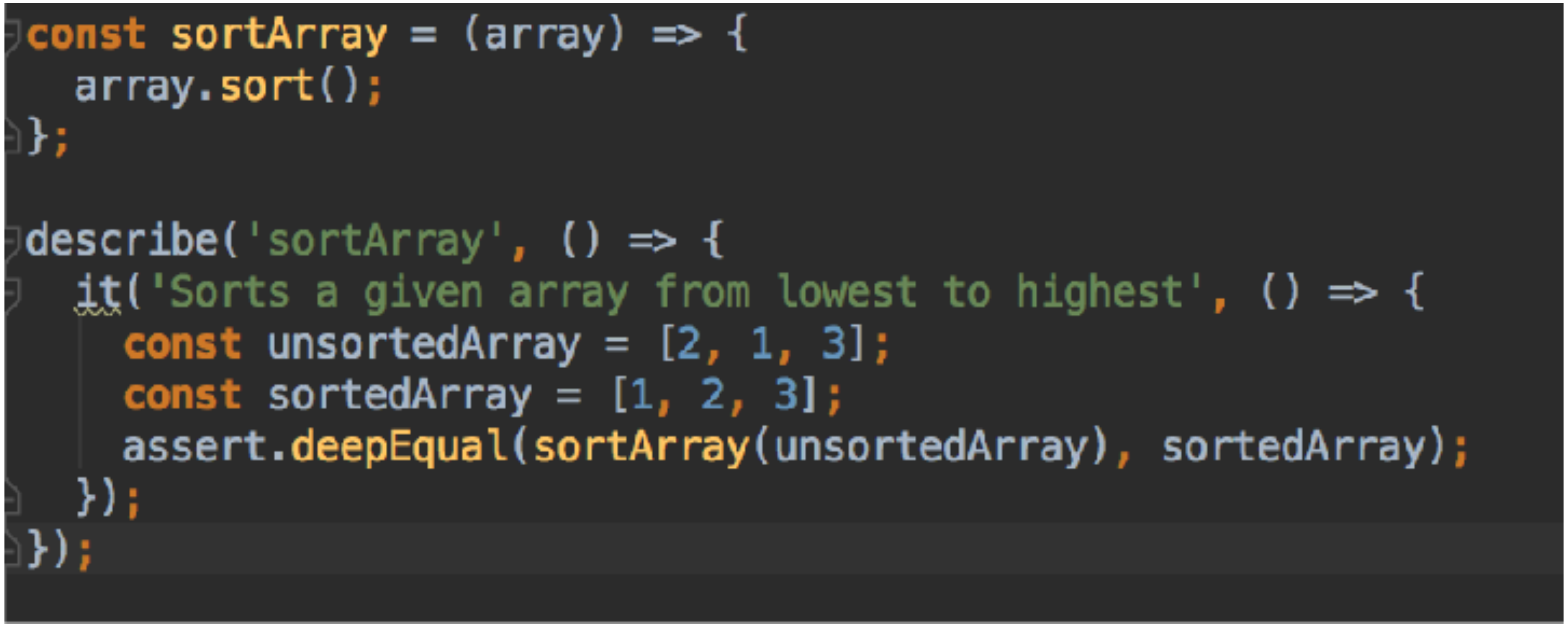
Questions

- Can you describe the test pyramid?
- Can you quantify the benefits or drawbacks of the various test types?
- Can you explain why we don't follow the test pyramid and where we should?
- Do you understand the difference between testing outcomes (yay!) vs interactions (boo)?

Notes 0

TESTING OUTCOMES

- We test that the code **returns** the right result

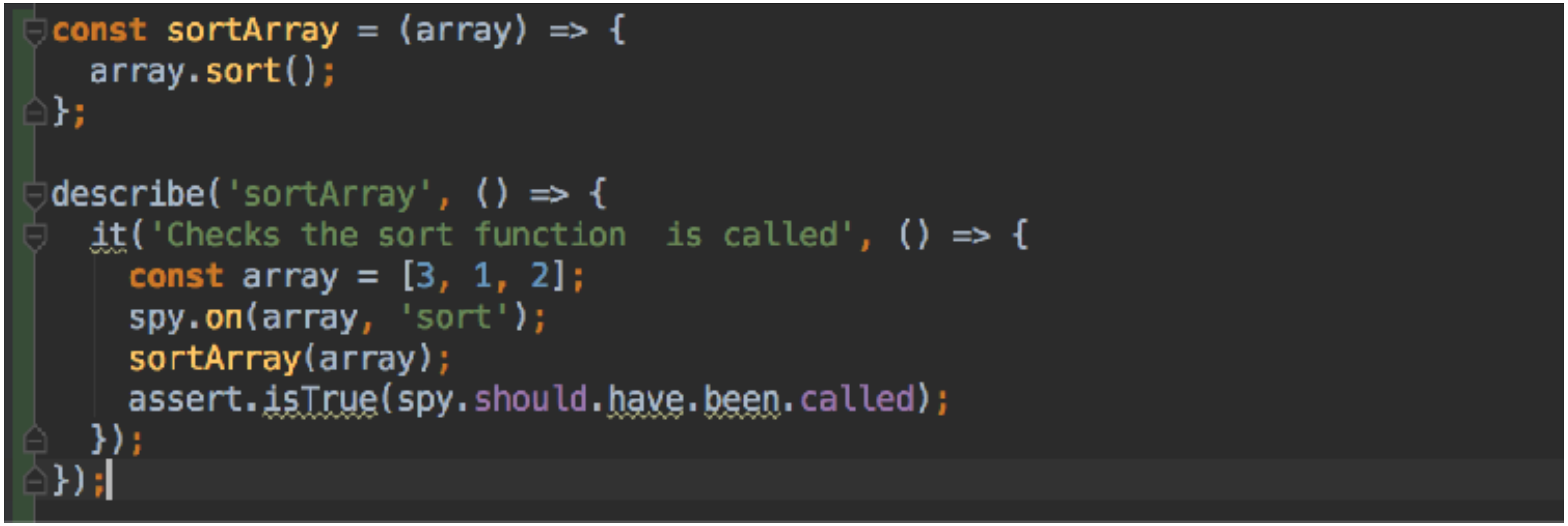


```
const sortArray = (array) => {
  array.sort();
};

describe('sortArray', () => {
  it('Sorts a given array from lowest to highest', () => {
    const unsortedArray = [2, 1, 3];
    const sortedArray = [1, 2, 3];
    assert.deepEqual(sortArray(unsortedArray), sortedArray);
  });
});
```

TESTING INTERACTIONS

- We test that the code **calls** certain functions properly.

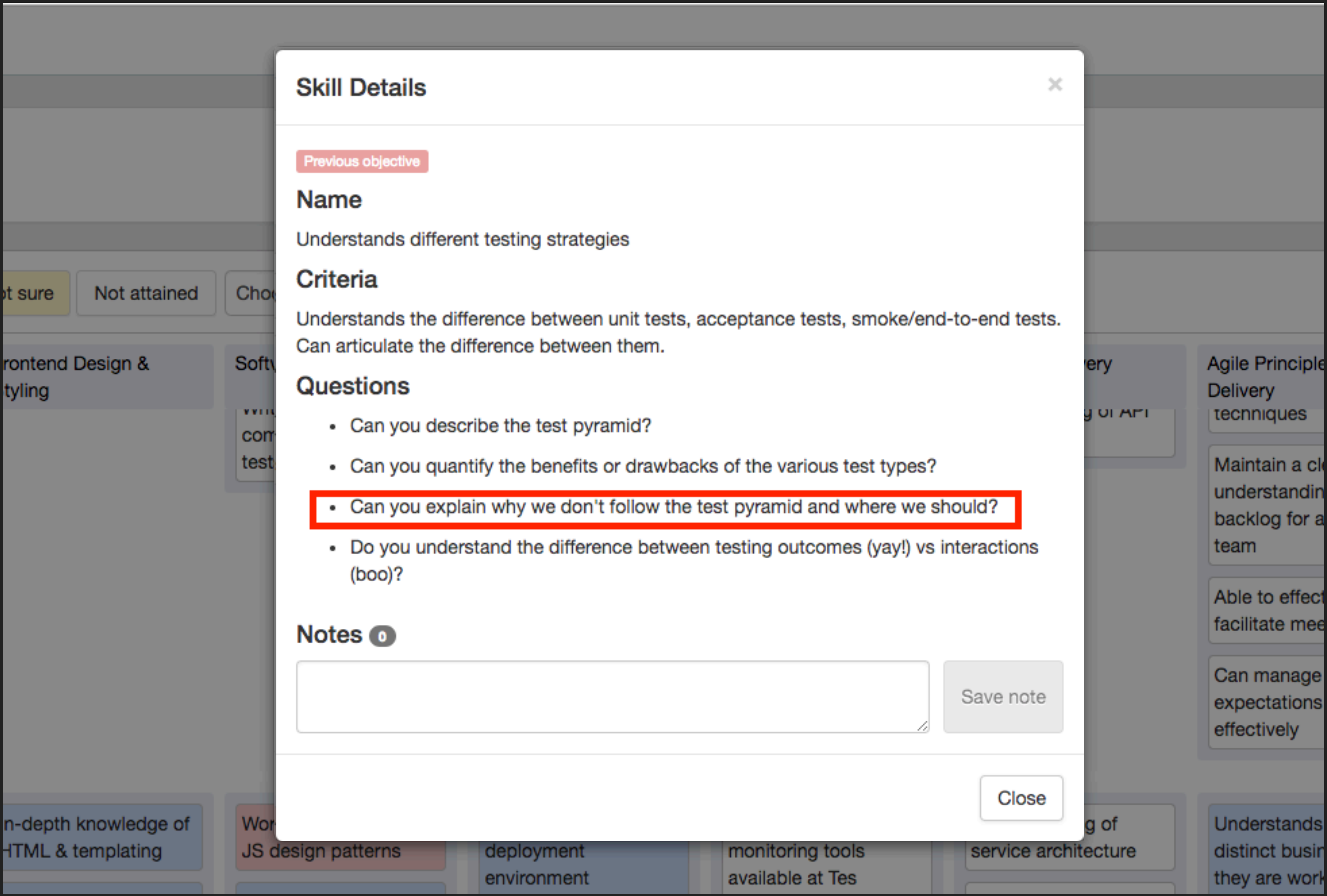


```
const sortArray = (array) => {
  array.sort();
};

describe('sortArray', () => {
  it('Checks the sort function  is called', () => {
    const array = [3, 1, 2];
    spy.on(array, 'sort');
    sortArray(array);
    assert.isTrue(spy.should.have.been.called);
  });
});
```

TESTING INTERACTIONS

- Gets good code coverage, but doesn’t tell you if the function called is actually working as expected.
- **Just because a test that uses interactions is passing doesn’t mean the code is working properly.**
- Can be useful in case where where the number of calls to a method is important (e.g. sending emails).



×
Previous objective
Save note
Close
t sure
Not attained
Cho
rontend Design & tyling
Soft
Agile Principle Delivery
techniques
Maintain a cl understandin backlog for a team
Able to effect facilitate mee
Can manage expectations effectively
n-depth knowledge of HTML & templating
Wor JS design patterns
deployment environment
monitoring tools available at Tes
g of service architecture
Understands distinct busi they are work

Skill Details

Name

Understands different testing strategies

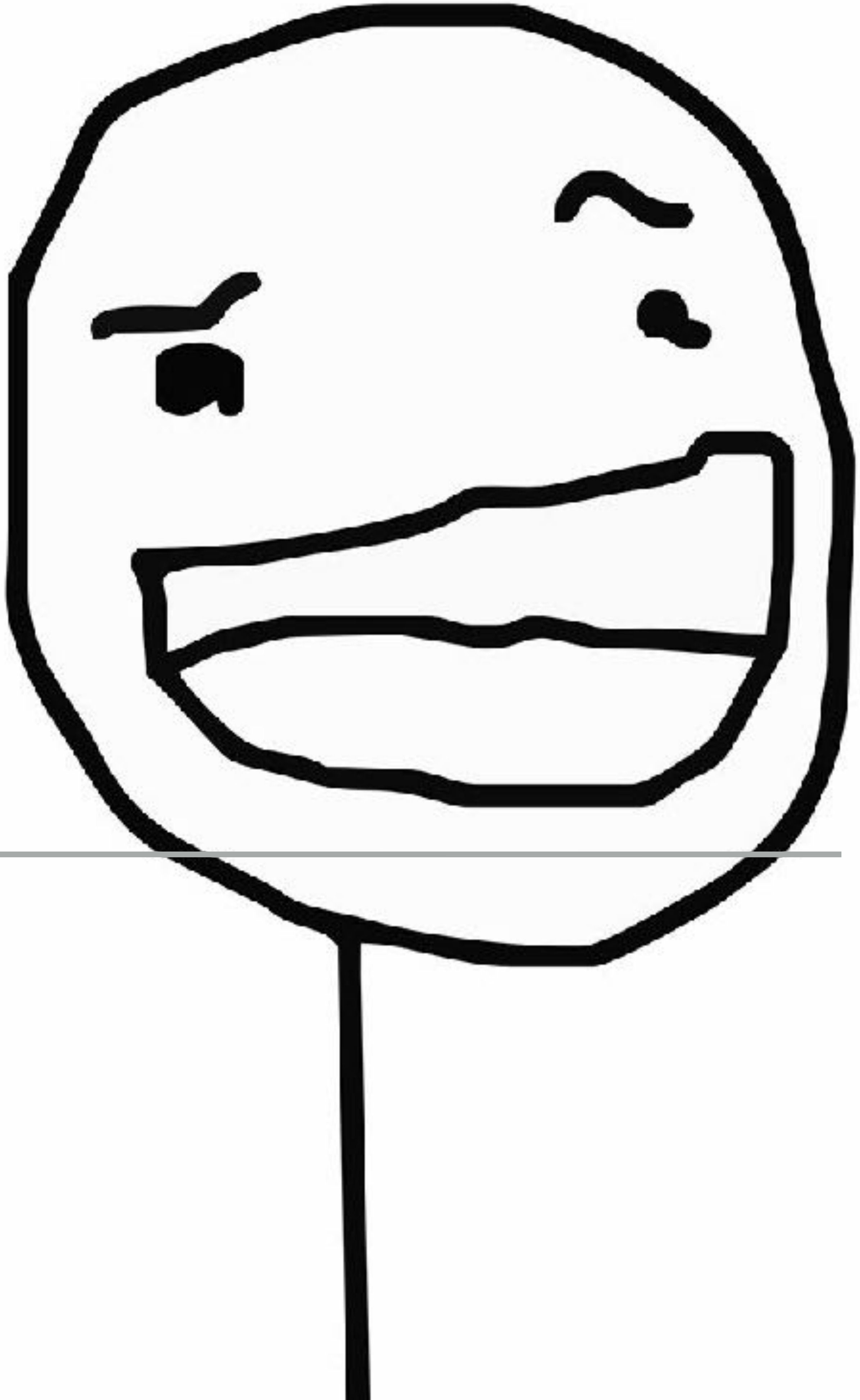
Criteria

Understands the difference between unit tests, acceptance tests, smoke/end-to-end tests. Can articulate the difference between them.

Questions

- Can you describe the test pyramid?
- Can you quantify the benefits or drawbacks of the various test types?
- Can you explain why we don't follow the test pyramid and where we should?
- Do you understand the difference between testing outcomes (yay!) vs interactions (boo)?

Notes 0



MAYBE?

SERVICE VS APP

- Tes architecture means there will always be a difference in the types of test required for each service or app.
- For example, we have a limited number of tests for app-marketing and no smoke tests, because it is an internal app, and so the costs outweigh the benefits.
- Broadly there seems to be lack of UI acceptance testing at Tes, as the costs are seen as too high vs reward.

ANY OTHER EXAMPLES?