

Application.Tests.UseCases.
Area.UpdateAreaTests.Execute
_WithNonExistingId_ShouldThrowException

Application.Tests.UseCases.
Area.UpdateAreaTests.Execute
_WithNullId_ShouldThrowException

Application.Tests.UseCases.
Area.UpdateAreaTests.Execute
_WithValidIdAndInput_ShouldReturnDetailed
ReadAreaOutput

Application.Interfaces.Use
Cases.Area.IUpdateArea.ExecuteAsync

```
graph LR; A["Application.Tests.UseCases.  
Area.UpdateAreaTests.Execute  
_WithNonExistingId_ShouldThrowException"] --> D["Application.Interfaces.Use  
Cases.Area.IUpdateArea.ExecuteAsync"]; B["Application.Tests.UseCases.  
Area.UpdateAreaTests.Execute  
_WithNullId_ShouldThrowException"] --> D; C["Application.Tests.UseCases.  
Area.UpdateAreaTests.Execute  
_WithValidIdAndInput_ShouldReturnDetailed  
ReadAreaOutput"] --> D;
```

The diagram illustrates a dependency or implementation relationship. Three test cases, each in a white box, have blue arrows pointing to a single interface method in a gray box. The test cases are for different input scenarios: non-existing ID, null ID, and valid ID with input. The interface method is `ExecuteAsync` on the `IUpdateArea` interface.