Testing (Flutter) Apps

CS 442: Mobile App Development

Michael Lee <lee@iit.edu>
Why should we write tests?
Types of tests

- Unit tests
- Component/Widget tests
- Integration tests
Unit test

• Tests a single function or class ("unit")

• Verify correct behavior with different input and conditions/state

  • May depend on external dependencies (e.g., file data, database)

• External dependencies may be problematic!

  • Slow down testing (e.g., http, remote database servers)

  • External exceptions may break tests (but units are not at fault)
“Mock” dependencies

• Instead of using real/live external dependencies, create mock versions
  • Emulate the live version, but without risk of slowdown/flakiness
Component/Widget test

• Tests a single UI component (Flutter Widget)
• Very correct visual and interactive aspects
• Requires simulating “lifecycle” events (e.g., building widget tree, performing layout) and user input / asynchronous events
• Flutter has comprehensive widget-testing facilities
Integration test

• Tests large portions of or the entire app

• Verify that all widgets, underlying logic, and external dependencies work together (integrate) correctly

• Recording behavior over test may be used to profile app performance

• Typically run in an emulator
## Trade-offs

<table>
<thead>
<tr>
<th></th>
<th>Unit</th>
<th>Widget</th>
<th>Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidence</td>
<td>Low</td>
<td>Higher</td>
<td>Highest</td>
</tr>
<tr>
<td>Maintenance cost</td>
<td>Low</td>
<td>Higher</td>
<td>Highest</td>
</tr>
<tr>
<td>Dependencies</td>
<td>Few</td>
<td>More</td>
<td>Most</td>
</tr>
<tr>
<td>Execution speed</td>
<td>Quick</td>
<td>Quick</td>
<td>Slow</td>
</tr>
</tbody>
</table>
Continuous Integration (CI)

• Service that runs tests automatically after each substantial code update
  • Typically on new commits/pushes/merges
• Critical part of DevOps!
Flutter testing packages

- **flutter_test**: the official testing library from Flutter
  - Unit and Widget testing utilities
- **test**: general-purpose Dart testing library
  - Unit tests for both pure Dart & Flutter
- **mockito**: object mocking library
- **integration_test**: for app-level testing
  - Multi-widget/screen tests; can also be used for profiling