

Automated GUI Testing, including Capture/Replay Tools

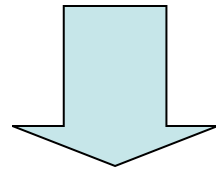
SWE 261P

Capture and Replay

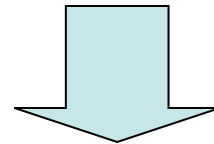
- Sometimes there is no alternative to human input and observation
 - Even if we separate testing program functionality from GUI, some testing of the GUI is required
- We can at least cut *repetition* of human testing
- *Capture* a manually run test case, *replay* it automatically
 - with a comparison-based test oracle: behavior same as previously accepted behavior
 - reusable only until a program change invalidates it
 - lifetime depends on abstraction level of input and output

Controllability & Observability

GUI input (MVC "Controller")



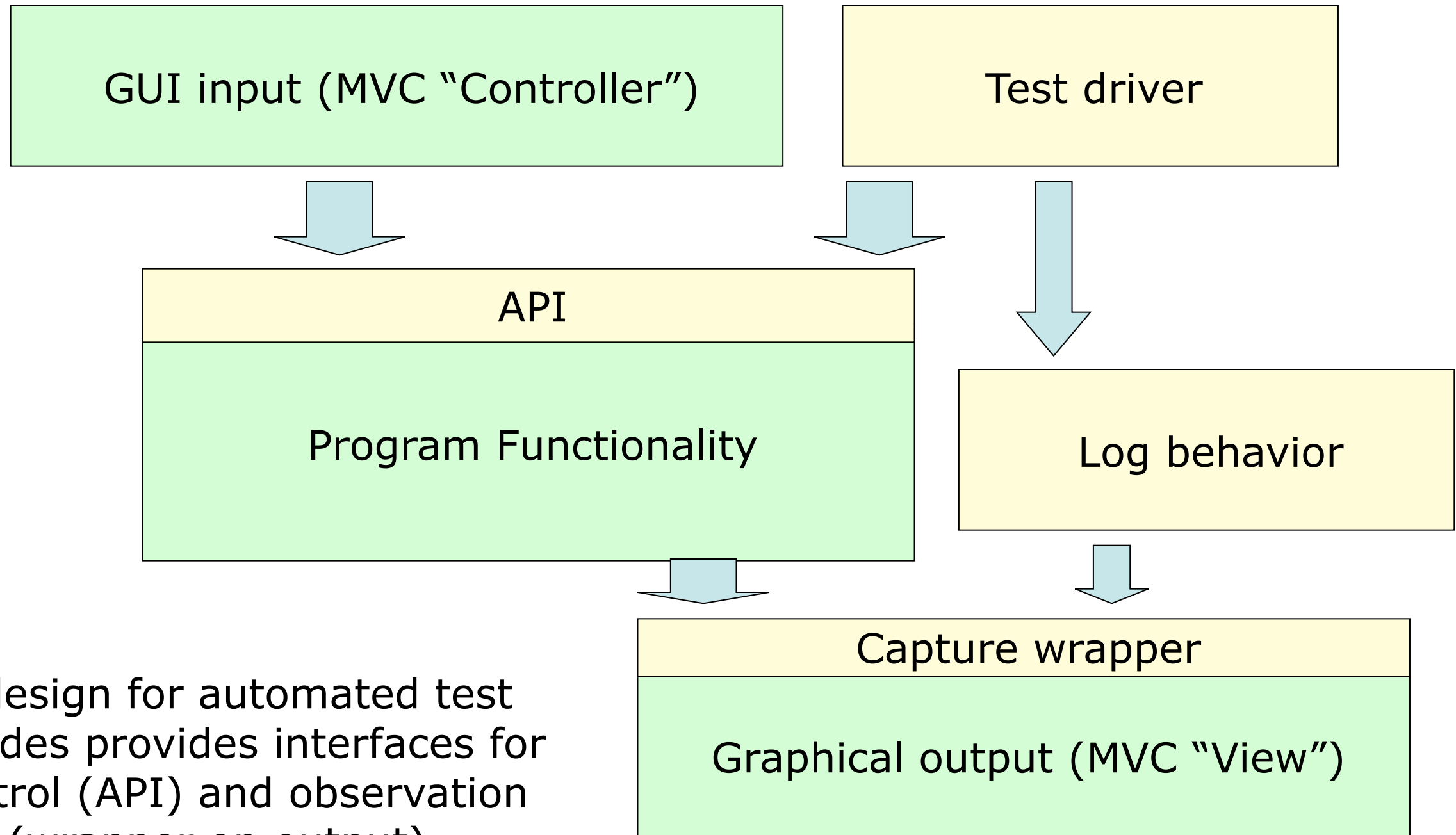
Program Functionality



Graphical output (MVC "View")

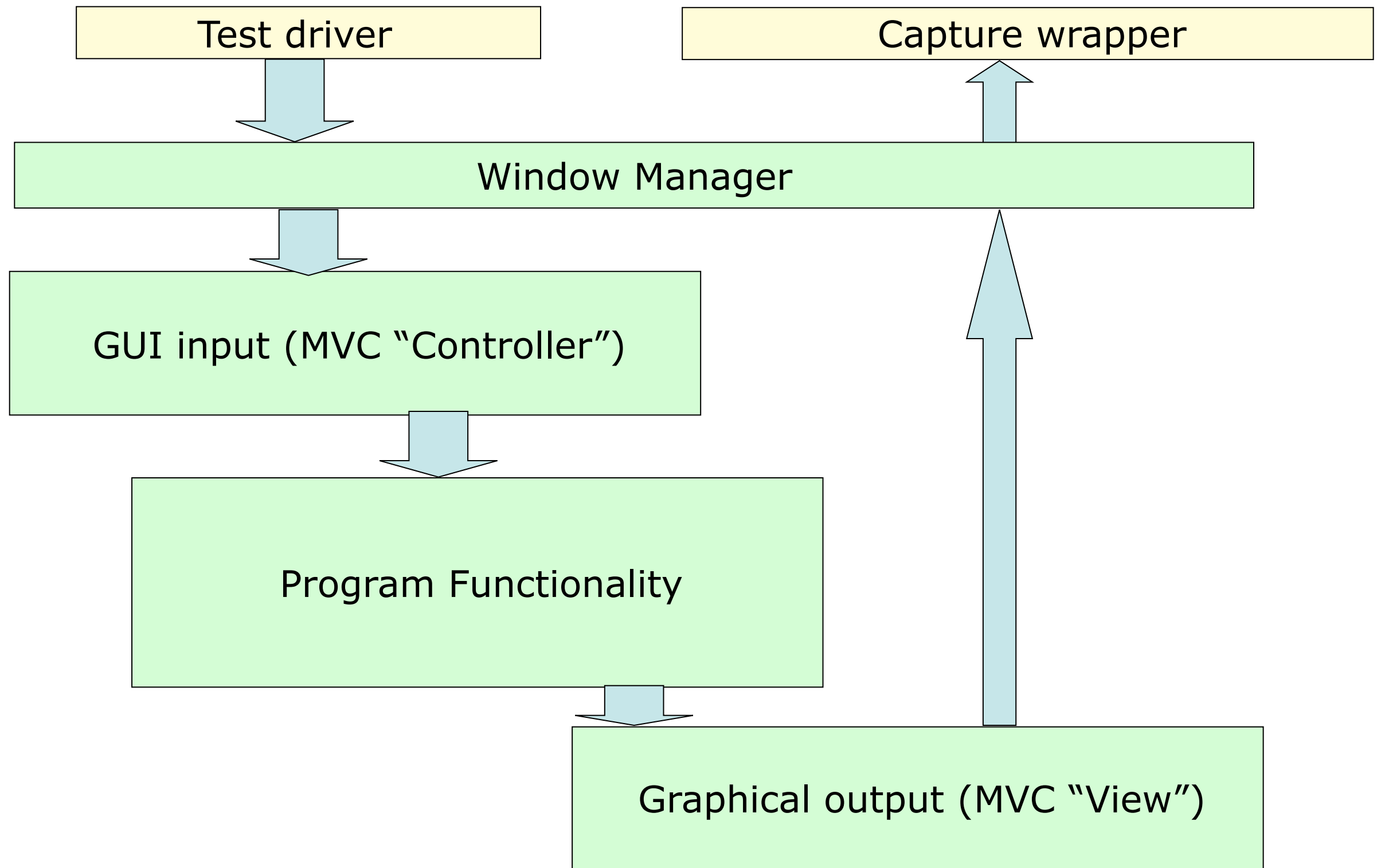
Example: We want automated tests, but interactive input provides limited control and graphical output provides limited observability

Controllability & Observability



A design for automated test includes provides interfaces for control (API) and observation (wrapper on output).

Generalizable but Brittle



Example

- Abbot Java GUI Test Framework:

```
// Suppose MyComponent has a text field and a button...
MyComponent comp = new MyComponent();
// Display a frame containing the given component
showFrame(comp);

JTextField textField = (JTextField)getFinder().
    find(new ClassMatcher(JTextField.class));
JButton button = (JButton)getFinder().find(new Matcher() {
    public boolean matches(Component c) {
        // Add as much information as needed to distinguish the component
        return c instanceof JButton && ((JButton)c).getText().equals("OK");
    }
});
JTextComponentTester tester = new JTextComponentTester();
tester.actionEnterText(textField, "This text is typed in the text field");
tester.actionClick(button);
// Perform some tests on the state of your UI or model
assertEquals("Wrong button tooltip", "Click here to accept", button.getToolTipText());
```

Example Capture/Replay Tools

- Ranorex Studio (Windows, object identification)
- Abbot (Java, API level)
- Selenium (Web browser, IDE, Java/C#/Python/more APIs)
- Jubula (Java, Swing, SWT/RCP/GEF and JavaFX, Windows and Linux)
- TestComplete (Desktop, web, mobile, object identification)
- ...

Complications to GUI Capture/ Replay

- Capture of pixels means layout changes can cause troubles
- All environment events can play a part in test behavior (mouse, keyboard, network, clock, hardware devices, database, ...)
- Timing can play a part in test behavior... but allowing replay at user speed would take time

Selenium Exercise

Go to <https://www.selenium.dev/downloads/>
And download the Selenium IDE for Chrome or Firefox.
Capture several test cases for Wikipedia.