Factoring your app for testability

Philippe Casgrain
Principal Developer
Lightspeed, Inc

Unit Testing and CI

- Unit Testing can be done with XojoUnit https://github.com/xojo/XojoUnit
- Continuous Integration can be done with Jenkins https://jenkins.io
- See « Unit Testing from Soup to Nuts » XDC 2016

The ideal function


Parameters
Function
Results

The ideal unit test

Test parameters

Function

Results

=?

Expected results

More test parameters

Function

More results

=?

More expected results

Unit tests are not System tests

Factoring TaskMaster

- TaskMaster has no real unit tests
- Let's remove the boilerplate tests and add our own

A quick primer on MVC

- Model: stores the data
- View: displays the data
  - Also allows user to update the data
- Controller: glues the two together

Model

- Represents an object from another domain in your application
  - Server endpoint, database result, image…
- Highly unit-testable
- Highly re-usable
  - The same model layer (and tests) can often be used across architectures (32 vs 64 bits, ARM vs Intel, …)

View

- Represents what the user sees and interacts with
  - Text entry boxes, buttons, scrolling lists, windows, menus…
- Rarely unit-testable
- Usually re-usable

Controller

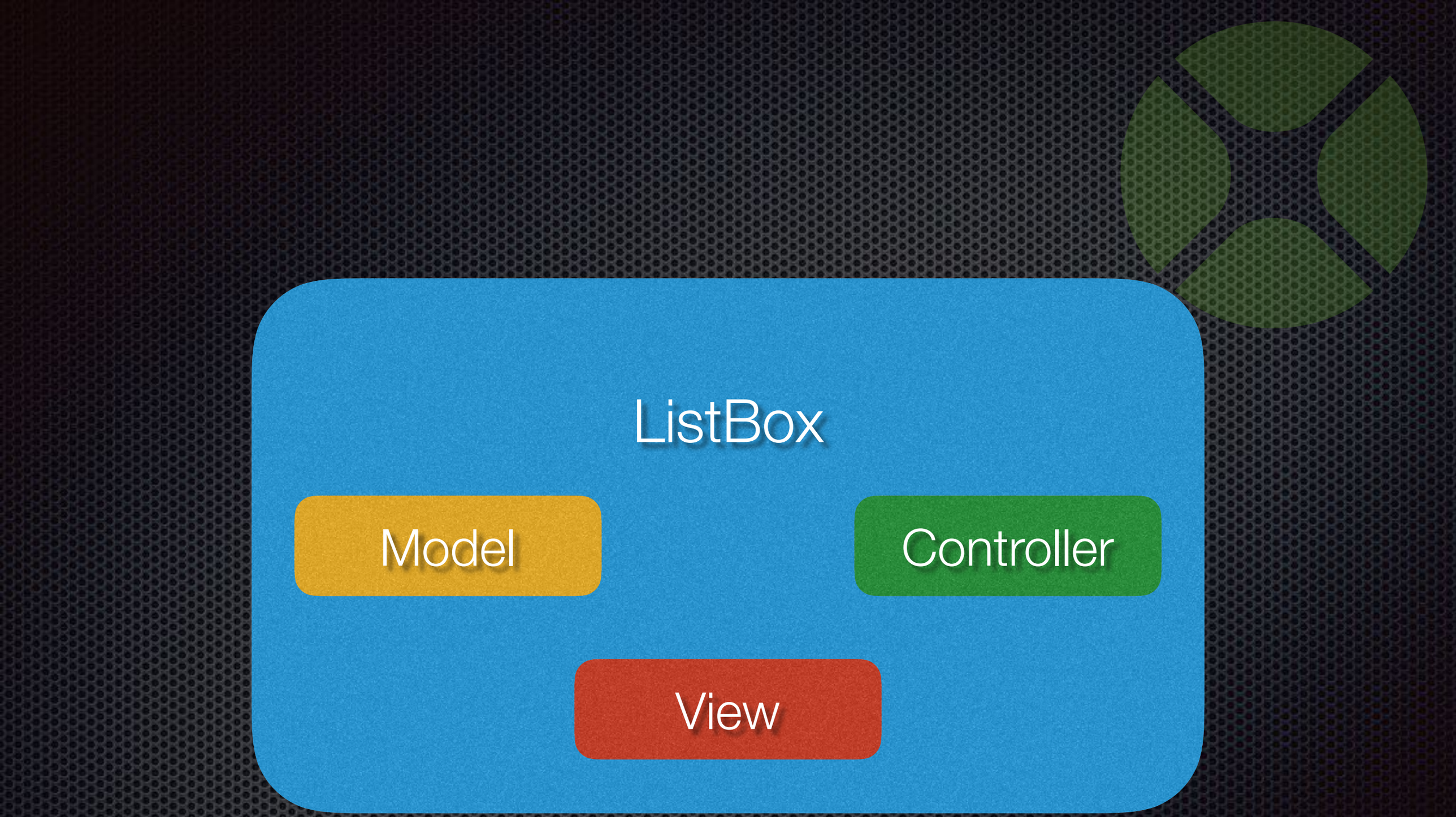
- Represents a mediator between your Views and your Models
- Somewhat testable
  - Lots of associated objects: **side-effects**
  - *Mock objects*
- Rarely re-usable

TaskMaster + Unit Tests

ListBox

- A model
- A view
- A controller

- Not easily testable


ListBox
Model
Controller
View

ListBox Refactoring

- Reacts to user input
- Knows how to display itself
- Needs a controller to interface with the model

App Refactoring

- Model
  - TaskItem
  - TaskList
- View
  - ListBox
- Controller
  - TaskController

TaskItem

- Simple representation with a Bool and a String
- Knows how to write itself to a string, and read itself from a string

TaskList

- Collection of TaskItems
- Knows how add, complete and delete tasks

TaskController

- Has a ListBox
- Has a TaskList
- Handles button events

ListBox Refactoring

Mock objects

Your friends when unit-testing

Mock Object

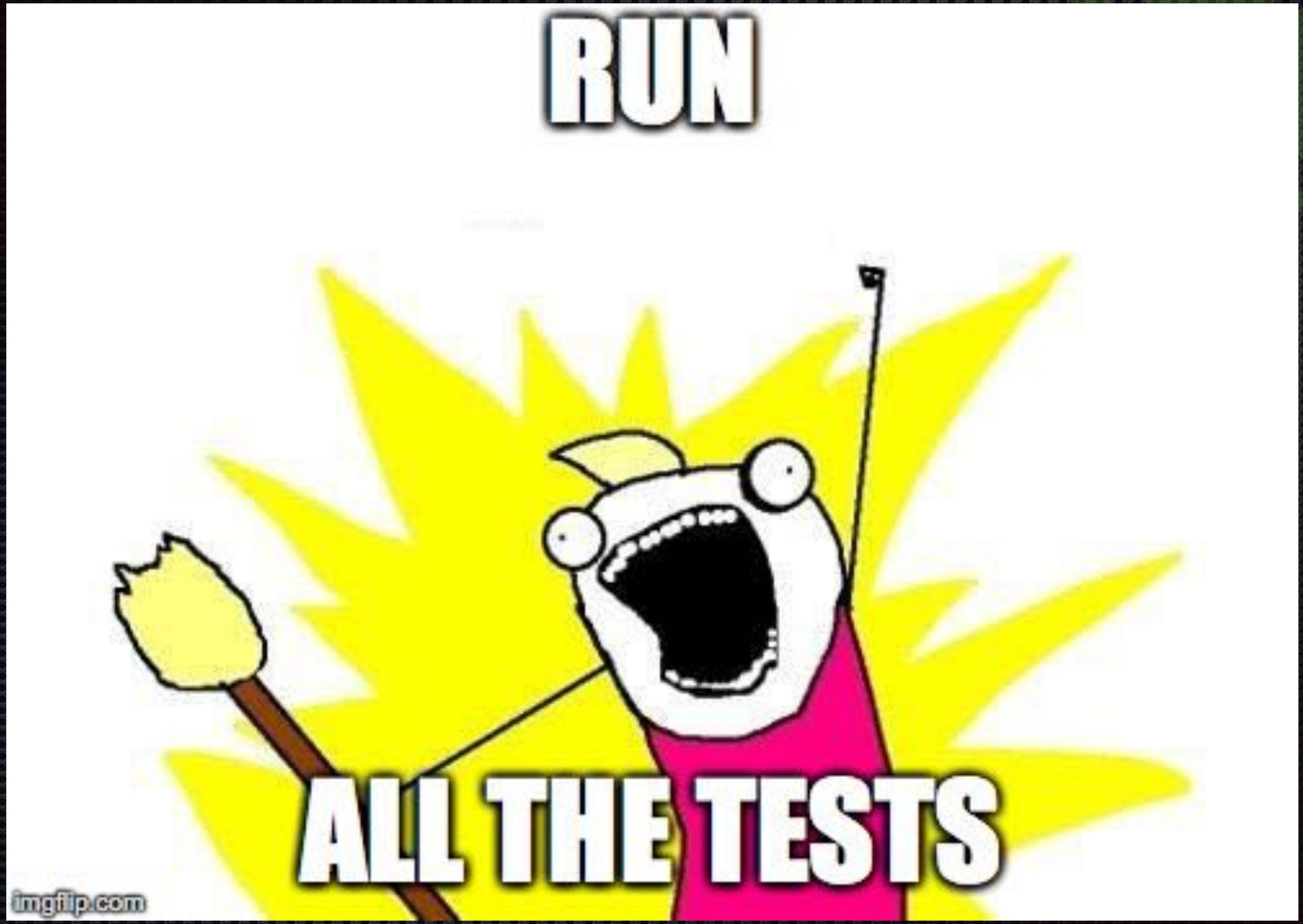
- Has the same interface as the object it mocks
  - Interface
  - Subclass
- Can return canned results
- Can set up expectations and verifications

TaskListInterface

- Create a new Interface with the methods from TaskList
  - « Extract Interface » is extremely useful
- Make TaskList conform to the Interface
- Change the property on the TaskController

TaskListMock

- New class that implements the Interface
- Returns canned data (or does nothing)
- Could be set up with expectations and verifications
- Used when unit-testing TaskListController


RUN
ALL THE TESTS
imgflip.com

Q & A

Philippe Casgrain

philippe.casgrain@lightspeedhq.com

Give us feedback on this session in the XDC app!

Resources

- https://github.com/philippec-ls/TaskMaster
- https://github.com/xojo/XojoUnit
- Martin Fowler, « Refactoring »