Testing and Go

The news

- Lets look and talk about it.

Another Survey

- How many of you have do automated testing?
- How many have used some sort of continuous integration?

Test Driven Development

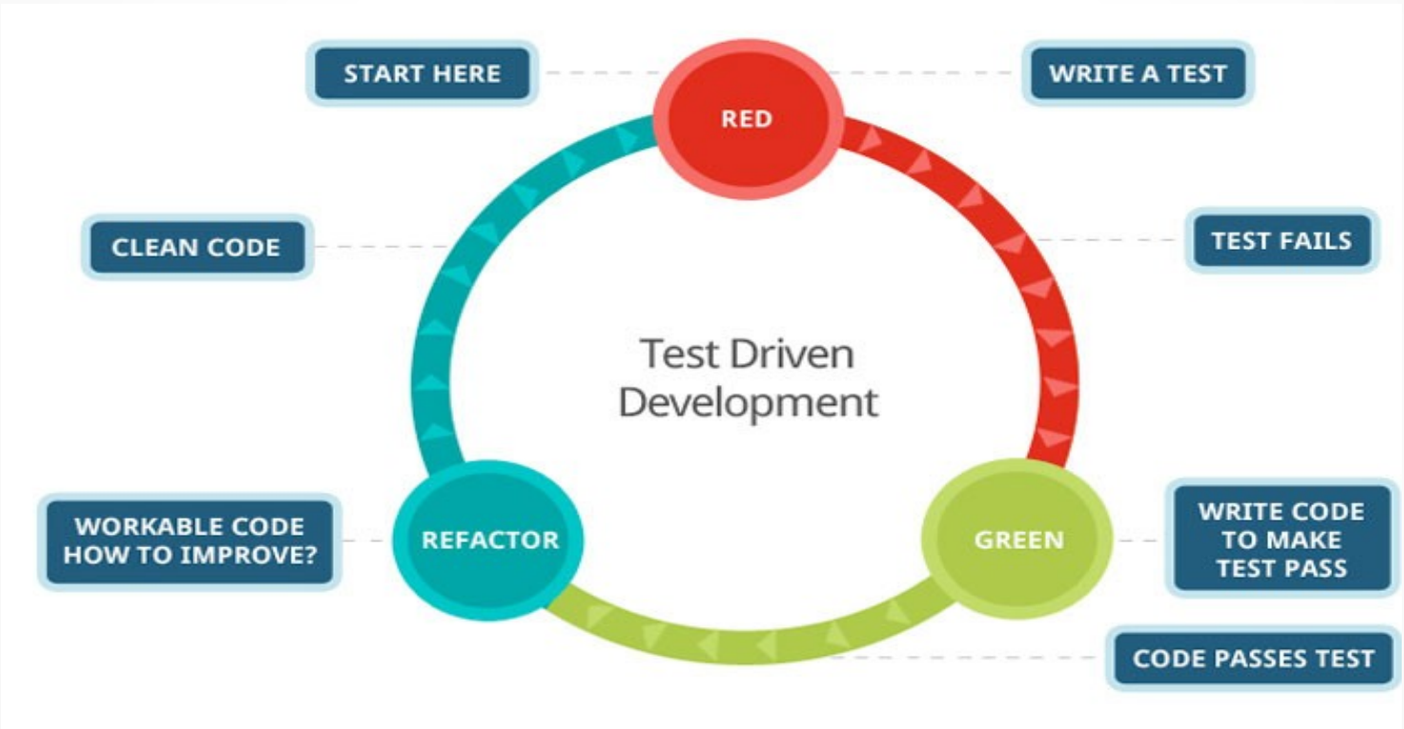
- Tests come first before you write code
  - Purist version?

Test Driven Development

- Tests come first before you write code
  - Purist version?
  - Write test before you do anything.
  - Want to write a webapp?
    - Before you do anything – including installing the web app libraries
      - Write a test.
      - When it fails do something.
  - So purist: write test, and only write real code when test fails.

Test Driven Development

- Purist approach.
-



Automated Tests

- So key is Automated tests
- So I keep hearing about these tests
  - What kinds of tests do we care about?
  - How do people categorize the different kinds of automated tests?
  - So many lucky volunteers

Automated Tests

- So I keep hearing about these tests
  - What kinds of tests do we care about?
    - Unit tests
    - Functional tests
    - Acceptance tests
    - What are each of these?

Automated Tests

- So I keep hearing about these tests
  - What kinds of tests do we care about?
    - Unit tests
      - Item by item – function by function tests
    - Functional tests
      - Does the app do what it is supposed to do?
    - Acceptance tests
      - Does the app do what the client thinks it is supposed to do?
  - Of course then we run into: what is an “item”?

Automated Tests

- So what are the tests supposed to do for us in Test Driven Development or (more and more often) other methods of using automated tests?
  - Why has Testing (TDD?) become so accepted in the last 10-15 years?
  - Well actually some people still call it TDD but ‘automated tests’ might be a better term
  - What does Automated testing buy us?

Automated Tests

- So what are the tests supposed to do for us in Test Driven Development or (more and more often) other methods of using automated tests?
  - Why has Testing (TDD?) become so accepted in the last 10-15 years?
  - Well actually some people still call it TDD but ‘automated tests’ might be a better term
  - Tests are run every time code is compiled/interpreted.
  - Tests become an extension of the compilers ability to catch errors.
  - Always better to let the compiler catch the error.
  - Why?
    - What does it buy us?

Tests for us

- I'm not a TDD person
  - But the automated tests technique are still valuable
  - Automated tests of some sort are more or less mandatory today.
  - Turn your specs into tests
    - Unit tests
    - And functional tests
    - Write them,
    - Then write the code
    - Then run the tests
    - Every time you change anything and build
      - Run all tests again

Unit Tests

- Testing Smallest Testable part of application
  - Functions, methods, etc
  - Sometimes the entire public interface to a class
  - Extend compiler's error checking capability.
- Traditionally each unit test should be done in isolation
  - Even if your class relies on a database, mock database and test class
  - Recently lots of conference talks pushing back against mocks, tests on each unit will include its dependancies
    - We'll see if this takes

Unit/Automated Tests

- Available in nearly every important language
  - JUnit (grandaddy of all unit tests – written by Kent Beck and Erich Gamma) – major updates with JUnit 5
  - Cheat for C
  - Googletest for c++
  - PyUnit for python(older library based on JUnit)
  - Pytest modern – can write tests without giant test classes
  - Built-in for newer languages like **go** and rust
  - etc

Testing Humor

Brenan Keller
@brenankeller

Follow

A QA engineer walks into a bar. Orders a beer. Orders 0 beers. Orders 99999999999 beers. Orders a lizard. Orders -1 beers. Orders a ueicbksjdhd.

First real customer walks in and asks where the bathroom is. The bar bursts into flames, killing everyone.

4:21 PM - 30 Nov 2018

Testing and bug bounties

- Why does every tech company that institutes an internal “bug bounty” program end it very quickly?
  - Lucky volunteer other than the people who took the capstone with me?

Testing and bug bounties

- Why does every tech company that institutes an internal “bug bounty” program end it very quickly?
  - Lucky volunteer other than the people who took the capstone with me?
  - 

Tests in Go

- Testing is built in to go
  - Put your tests into a file with a name that ends in _test.go
  - Eg
    - For the databaseDemo.go I might have
    - databaseDemo_test.go
  - Import the testing package in each test file

Test Functions

- Test functions need to be exported symbols which means they start with?
  - Lucky volunteer?

Test Functions

- Test functions need to be exported symbols which means they start with?
  - A capital letter
- All tests functions start with Test
  - Rest of function name should start with capital letter too
  - Eg
    - TestOpenDataBase(t *testing.T)

Errors and the “Unix Way”

- The old “Unix way” developed in the days of command line terminals
  - Everything works: no feedback
  - Something went wrong: show error feedback
  - Use t.Error(<error message>) to fail the test

Running tests

- Using command line tools:
  - go test XXX_test.go
    - Will run tests just in that file
  - go test
    - Will run all tests in the current directory.

Table Tests

- One of the neat go testing mechanisms is “table testing”
- At the beginning of the testing function
  - Create a struct with
    - Test data and then final item is expected output
  - Create a slice of literals for the struct
  - For loop through the slice and apply the tests and check the outcome for each.

Example

```
func TestAbs(t *testing.T) {
    tests := []struct {
        input    int
        expected int
    }{
        {1, 1},
        {0, 0},
        {-1, 1},
        {-maxInt, maxInt},
        {maxInt, maxInt},
    }
    for _, test := range tests {
        actual := Abs(test.input)
        if actual != test.expected {
            t.Errorf("Abs(%d) = %d; want %d", test.input, actual, test.expected)
        }
    }
}
```

Example from:
https://lwn.net/Articles/821358/

Now lets try some

- Lets go write some tests.

Continuous Integration

- What do we mean by continuous integration?

Continuous Integration

- What do we mean by continuous integration?
  - Every time we commit code to version control, the entire project is built and tested.
  - Compare to previous approaches
    - Group might work on its piece of the project, maybe a library, and build and test it in isolation except for occasional “gold master” style builds
  - Now, since automated tests run for every commit/push/pull request,
    - you are either fairly confident that the new changes don’t break the existing project
    - Of find out about the breaks right away.

Experiences

- Has anyone worked with Continuous Integration before?
  - What sorts?
  -
  -

Experiences

- Has anyone worked with Continuous Integration before?
  - What sorts?
    - Jenkins
    - TravisCI
    - CircleCI
    - Jetbrains TeamCity
    - CodeShip
    - Bamboo
    - etc

Continuous Integration

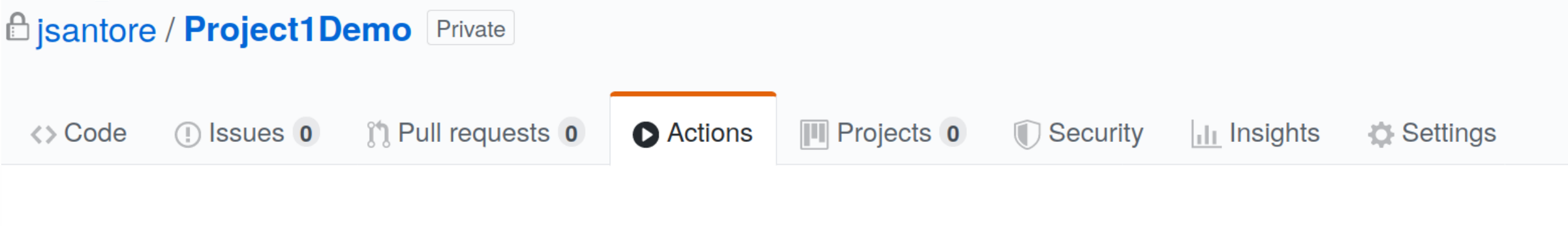
- Today the top two cloud based git servers provide CI services too
  - Gitlab has had CI for years
  - Github introduced github actions a few years ago
    - And made them free in August 2019 for everyone.
    - We use github for this class
      - Since the jetbrains integration with github is really good.
      - And the gitlab integration is only so-so

Lets try it out

- Lets pick a go repository and add github actions

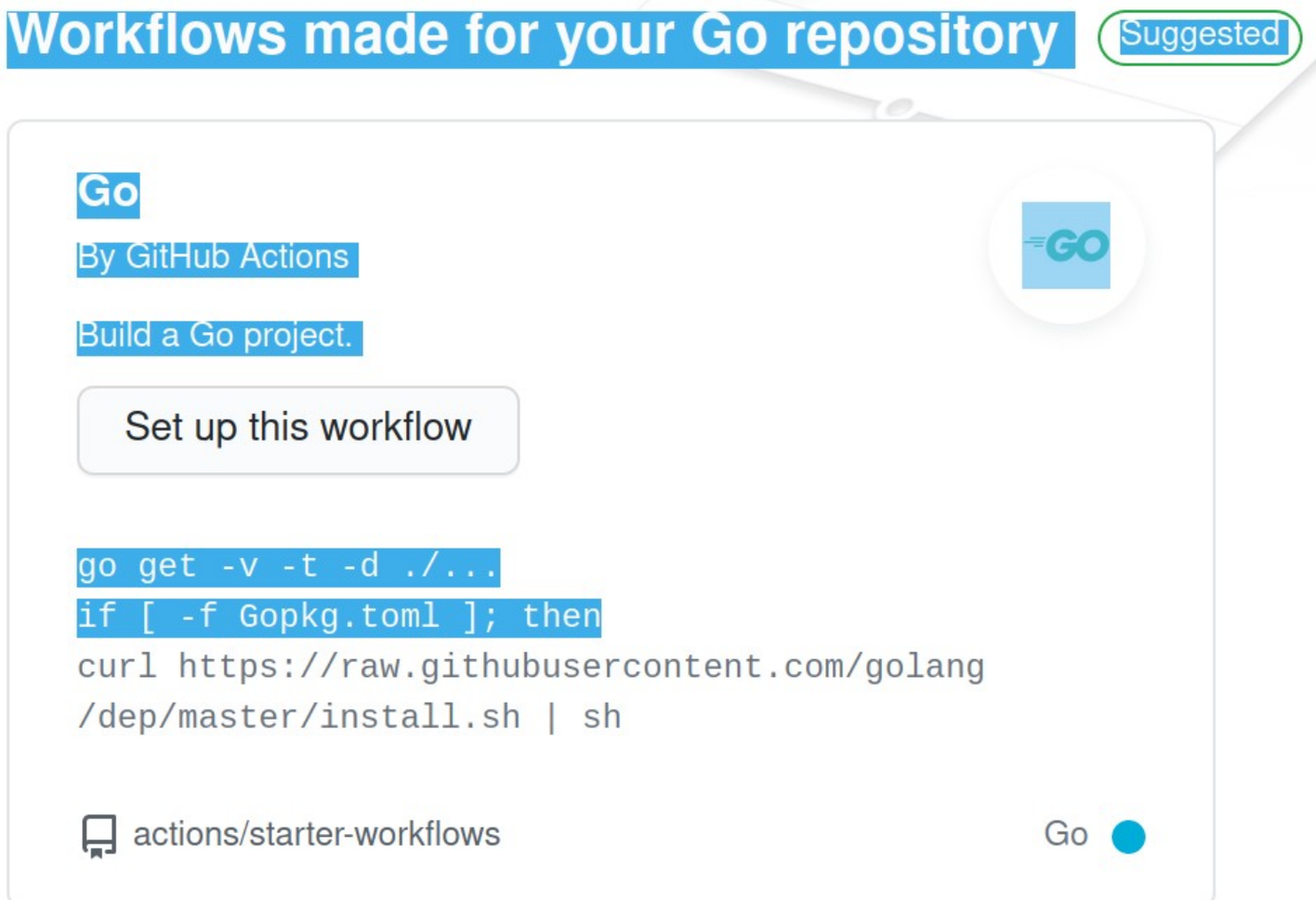
Adding Actions

- First Click the Actions Tab



Go

- There is one prebuilt for go projects
-



Go default

```
# This workflow will build a golang project
# For more information see: https://docs.github.com/en/actions/automating-builds-and-tests/building-and-testing-go

name: Go

on:
  push:
    branches: [ "master" ]
  pull_request:
    branches: [ "master" ]

jobs:

  build:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v3

    - name: Set up Go
      uses: actions/setup-go@v4
      with:
        go-version: '1.20'

    - name: Build
      run: go build -v ./...

    - name: Test
      run: go test -v ./...
```

- This is small, lets look at it on github itself.
- Note gofmt is no longer run
- Assumes you have done so
- But tests are run which makes the valuable.
- If tests fail – all collaborators get screaming email

Now lets make a commit

- And see the results.

Lets call it here for now