Drew Dolgert github@dolgert.com
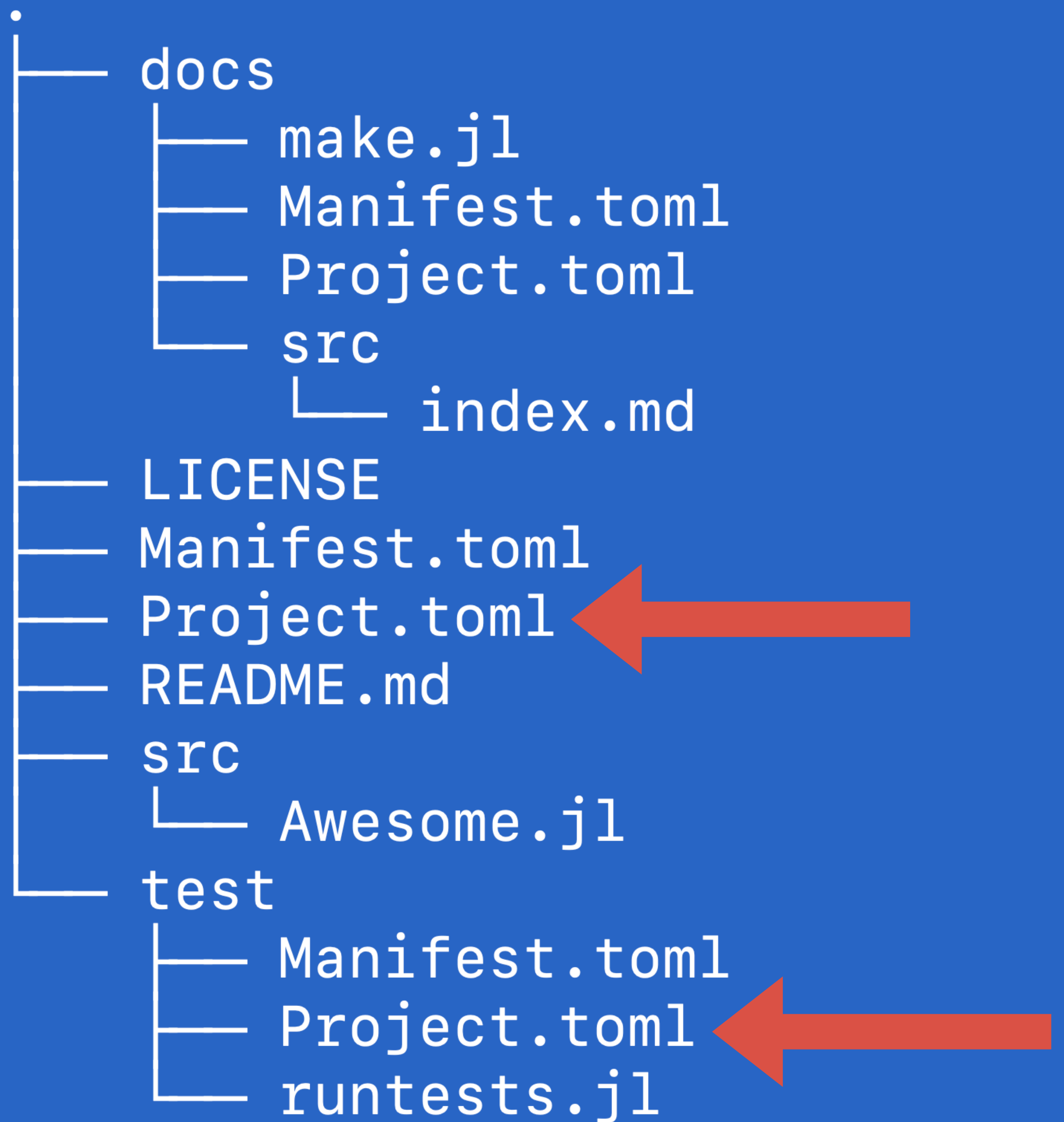
Julia Testing Packages

```
using PkgTemplates

t = Template()

t("Awesome")
```

```
.
├── docs
│   ├── make.jl
│   ├── Manifest.toml
│   ├── Project.toml
│   └── src
│       └── index.md
├── LICENSE
├── Manifest.toml
├── Project.toml          ⬅
├── README.md
├── src
│   └── Awesome.jl
└── test
    ├── Manifest.toml
    ├── Project.toml       ⬅
    └── runtests.jl
```

Review Criteria

- JuliaHub.com & <u>JuliaPackages.com</u> - "test"

- Project.toml[test=] & test/Project.toml - All 12,000 packages and 1,000 Github w/"test".

- Not using the live feed of package downloads, but I've scanned it since.

- Exclude if won't install on LTS 1.10.

- Exclude if specific to application domain (eg fluid dynamics, nail polish colors).

- Tracked in github.com/adolgert/JuliaTestDeps.jl.

Julia packages used in the test/ directory aren't reported in JuliaHub statistics.

CMU Software Engineering Institute

- Federally-funded research and development center (FFRDC)

- Three divisions: CERT, Software Solutions Division, AI Division (I'm here)

Dolgert:

- Julia since v0.3, 3 packages: UnitTestDesign.jl, BijectiveHilbert.jl, CompetingClocks.jl.

- Probability and stochastic simulation, AI, HPC, demography, physicist.

- Testing *scientific code*.

Organization: Pretend We're Testing

1. Choose a system-under-test or function-under-test.

2. Generate test cases.

3. Select test cases.

4. Run test cases.

5. Report test case outcomes.

International Software Testing Qualifications Board, istqb.org

In test/ But Not (Exactly) Testing

- Weave.jl & Literate.jl - Report generation/literate programming.

- PackageCompiler.jl - Create system image for a Julia package.

- Infiltrator.jl - drop into live debugger!

- Revise.jl (edit package without restarting)

- LiveServer.jl (live-reload web server)

Test All The Things

- DocInventories.jl - Testing the documentation

- Docs.undocumented_names()

- Benchmark

  - TimerOutputs.jl

  - BenchmarkTools.jl

  - PkgBenchmark.jl

  - ChairMarks.jl

  - AllocCheck.jl - test memory allocations!

Testing the Package Itself

- InteractiveUtils.jl in Stdlib - to ensure what is exported, check globals, check correct function called, and more.

- Aqua.jl - Aqua.test_all(YourPackage)

- JET.jl - @test isempty(JET.get_reports(report_package(YourPackage)))

- IntegrationTests.jl - Uses CI/CD to test that packages behave together.

Test A Specific Interface

- REST API - BrokenRecord.jl (Record HTTP and play back)

- GUI - ImGuiTestEngine.jl (Dear ImGui windowing library)

- Terminal UI - TerminalRegressionTests.jl (Uses VT100.jl package)

Find Tests in Other Places

- README.md - TestReadme.jl

- Documentation - Documenter.jl: ```` ```jldoctest``` ````

- Notebooks - PlutoTest.jl, NBInclude.jl

- Latex-ish Documents - NarrativeTest.jl: Markdown format.

- Finding tests within the package - Later…

Transfer Files for Testing

- Downloads.jl - download()

- DataDeps.jl - Create, retrieve data by UUID from installation-dependent directories.

- RemoteFiles.jl - Download and check for updates.

- Pkg.Artifacts.jl - Installable binary dependencies.

- AWSS3.jl

- GoogleDrive.jl

JLD2

SQLITE

NETCDF

ZIPFILE

XLSX

ARROW

SERIALIZATION

JSON3

FILEIO

EZXML

HDF5

TABLES

TAR

NPZ

NPZ

BSON

Fixtures

- Mocking.jl - temporarily overload function calls for testing.

- Suppressor.jl, IOCapture.jl, LoggingExtras.jl - to capture stderr, stdout.

- Scratch.jl - to write temporary files.

- DotEnv.jl - to create sets of environment variables.

- StableRandom.jl - to make random numbers the same across Julia versions.

- Reexport.jl - to make private members available

Property-based Testing

- Supposition.jl

- RandomizedPropertyTest.jl

- PropCheck.jl

```
Supposition.@check function commutative(a=intgen, b=intgen)

    add(a,b) == add(b,a)

end

Supposition.@check function failprop(x=intgen)

    add(x, one(x)) < x

end
```

Combinatorial Testing

- With Test.@testset `@testset ["description $v, $w"] for v in itrv, w in itrw test_ex end`

- UnitTestDesign.jl

- BranchTests.jl

```julia
julia> test_cases = all_pairs(
           [1, 2, 3], ["low", "mid" ,"high"], [1.0, 3.7, 4.9], [:greedy, :relax, :optim])
10-element Vector{Vector{Any}}:
 [1, "low", 1.0, :greedy]
 [1, "mid", 3.7, :relax]
 [1, "high", 4.9, :optim]
 [2, "low", 3.7, :optim]
 [2, "mid", 1.0, :greedy]
 [2, "high", 1.0, :relax]
 [3, "low", 4.9, :relax]
 [3, "mid", 1.0, :optim]
 [3, "high", 3.7, :greedy]
 [2, "mid", 4.9, :greedy]
```

```julia
@testbranch "Vector" begin
    v = Vector{Int}()
    @test isempty(v)

    @testbranch "adds one element" begin
        push!(v, 1)
        @test length(v) == 1

        @testbranch "adds another element" begin
            push!(v, 2)
            @test length(v) == 2
        end

        @testbranch "removes element" begin
            pop!(v)
            @test isempty(v)
        end
    end
end
```

Testing by Comparison

- Other implementations - RCall, PythonCall, MAT(LAB), libdl.

- Recorded files - ReferenceTests.jl

- Other data types

  - Symbolics, DoubleFloats, FixedPointNumbers, IntervalArithmetic, SymPy

  - SparseArrays, OffsetArrays, Unitful, PooledArrays, Strided

- Other Plots: VisualRegressionTesting.jl

Improved Assertions

- Safetestsets.jl - put test sets into a module.

- Base - @inferred to check for type stability.

- DeepDiffs.jl - Walks into structs and lists.

- StructEquality.jl - Compares through struct members.

- MacroTools.jl - Inspect the Expr-form of a function.

- IntervalSets.jl, DomainSets.jl, Intervals.jl - Use floating-point intervals to express assertions.

- Unitful.jl - Check that units are correct (known as metamorphic testing).

- Jive.jl - A bunch of helpers.

The Deal with Test.jl

- The default Julia Test Framework.

- Runs tests as it finds them.

- Test code is in test/ because code inside a package would slow loading times.

Test Frameworks of the Past

- - FactCheck, 9 years ago

- - Jig, 12 years ago

- - JulieTest, 9 years ago

- - PyTest, 7 years ago

- - RunTests, 9 years

- - Saute, 13 years ago

Tiny Test Frameworks

- AssociatedTests.jl - tests next to functions they test.

- NestedTests.jl - for sharing setup/teardown of tests.

- Pukeko.jl - minimal, function-based, parallel test execution.

- TidyTest.jl - works with Test.jl and adds discovery of tests and test selection.
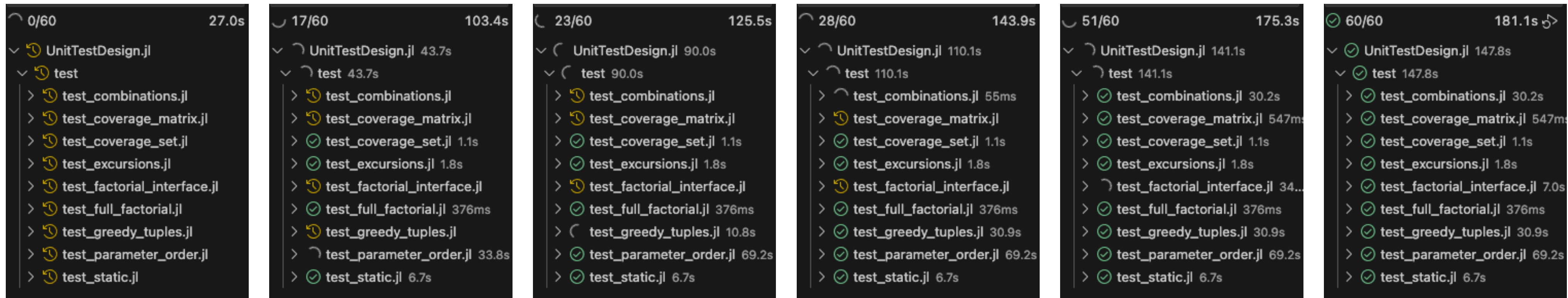
Test Frameworks

- TestItems.jl, TestItemRunner.jl, ReTestItems.jl - Removes test code from package, run in VSCode.

- ReTest.jl - collect tests, select, run in parallel.

- XUnit.jl - collect tests, select, schedule, run.

- Jute.jl - collect tests, parametrizes with fixtures.

Help Running Unit Tests

- TestTools.jl - CLI tools for testing, coverage, style checking.

- EyeOfRa.jl - Live rerunning!

- TestPicker.jl - REPL mode for running tests.

Blinky Green Lights: Test Output

- VisualStudio Plugin

- DotTestSets - Output like Python unittest. Watch the dots!

- Maracas - "Syntactic sugar and verbose output"

- TestSetExtensions - Readable output, DeepDiffs, and more within Test framework.

- On Github/Gitlab, other through XML file.

Test Quality

- Coverage.jl, LocalCoverage.jl - Count lines exercised by tests.

- Vimes.jl - Mutation testing (doesn't install on LTS 1.10).

- Other kinds of coverage? Branch, function, path, modified condition/decision coverage?

What's Missing?

- Test selection - Like pytest and pytest-testmon.

- Package to help make randomized tests - probabilistic programming to search for errors.