DrTests

-

The future of testing in Pharo

Julien Delplanque

julien.delplanque@live.be

Developers

Julien

- Entering last year of PhD in RMoD team
- Hacking Pharo around many aspects… just for fun :-)
- Hit by testing topic « by accident »

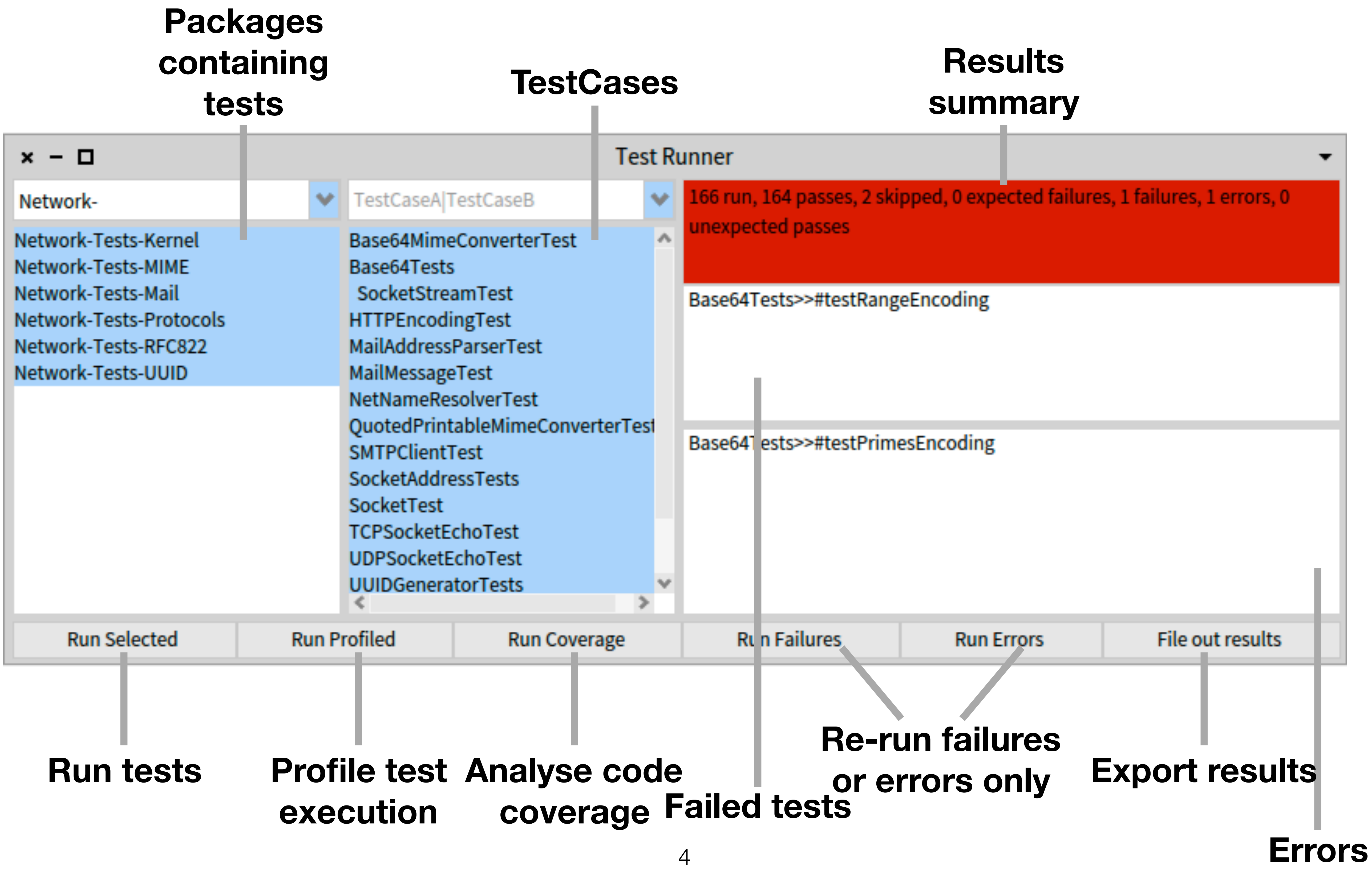
Dayne

- 2nd year of Master
- Did an internship in RMoD team working on DrTests for the last 6 months
- Participated to GSOC

Roadmap

- SUnit’s TestRunner and its limitations
- DrTests - an architecture to build tools around tests
- DrTests internal
- Next steps

TestRunner UI



Network-Tests-Kernel
Network-Tests-MIME
Network-Tests-Mail
Network-Tests-Protocols
Network-Tests-RFC822
Network-Tests-UUID

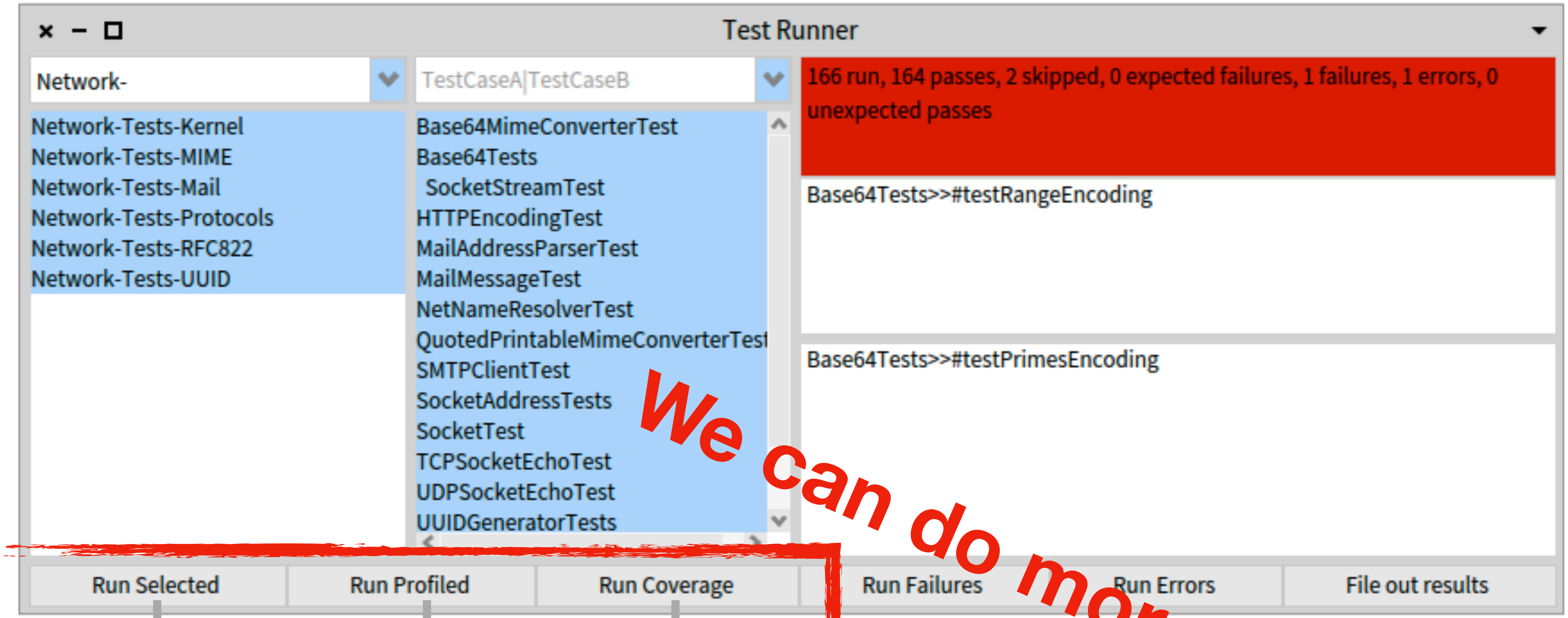
Base64MimeConverterTest
Base64Tests
SocketStreamTest
HTTPEncodingTest
MailAddressParserTest
MailMessageTest
NetNameResolverTest
QuotedPrintableMimeConverterTest
SMTPClientTest
SocketAddressTests
SocketTest
TCPSocketEchoTest
UDPSocketEchoTest
UUIDGeneratorTests

166 run, 164 passes, 2 skipped, 0 expected failures, 1 failures, 1 errors, 0 unexpected passes

Base64Tests>>#testRangeEncoding

Base64Tests>>#testPrimesEncoding

TestRunner UI



Network-Tests-Kernel
Network-Tests-MIME
Network-Tests-Mail
Network-Tests-Protocols
Network-Tests-RFC822
Network-Tests-UUID

Base64MimeConverterTest
Base64Tests
SocketStreamTest
HTTPEncodingTest
MailAddressParserTest
MailMessageTest
NetNameResolverTest
QuotedPrintableMimeConverterTest
SMTPClientTest
SocketAddressTests
SocketTest
TCPSocketEchoTest
UDPSocketEchoTest
UUIDGeneratorTests

166 run, 164 passes, 2 skipped, 0 expected failures, 1 failures, 1 errors, 0 unexpected passes

Base64Tests>>#testRangeEncoding

Base64Tests>>#testPrimesEncoding

Run tests

Profile test execution

Analyse code coverage

We can do more than that!

What can we do around tests?

Run them!

Test profiling

Test coverage

Parametrisable tests

Find rotten green tests

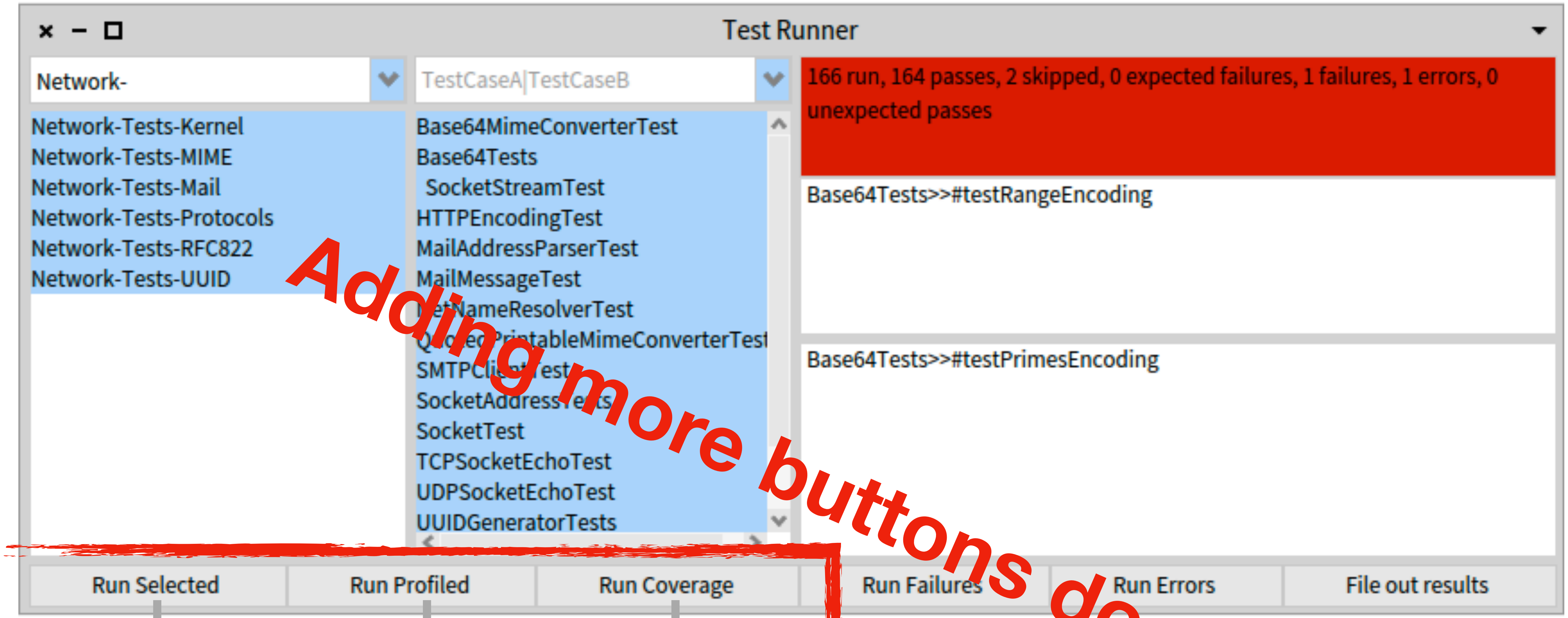
Mutation testing

Check executable comments

⋮

and more!

TestRunner UI



Test Runner

Network-

TestCaseA|TestCaseB

Network-Tests-Kernel
Network-Tests-MIME
Network-Tests-Mail
Network-Tests-Protocols
Network-Tests-RFC822
Network-Tests-UUID

Base64MimeConverterTest
Base64Tests
SocketStreamTest
HTTPEncodingTest
MailAddressParserTest
MailMessageTest
NetNameResolverTest
QuotedPrintableMimeConverterTest
SMTPClientTest
SocketAddressTests
SocketTest
TCPSocketEchoTest
UDPSocketEchoTest
UUIDGeneratorTests

166 run, 164 passes, 2 skipped, 0 expected failures, 1 failures, 1 errors, 0 unexpected passes

Base64Tests>>#testRangeEncoding

Base64Tests>>#testPrimesEncoding

Run tests **Profile test execution** **Analyse code coverage**

Adding more buttons does not scale!

TestRunner model

What do we want?

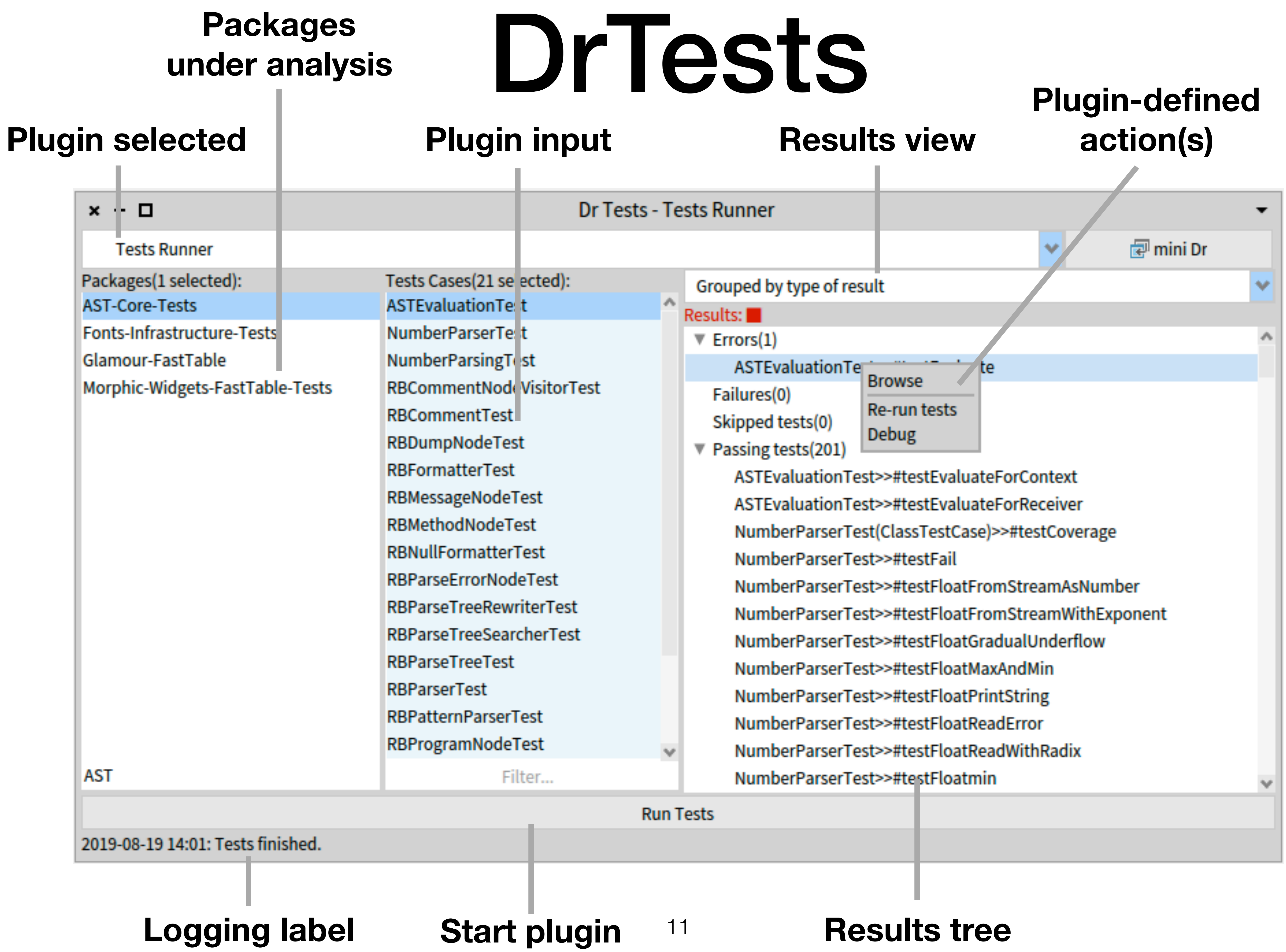
Power-up testing experience in Pharo by:

- Letting you plug **your analysis** via plugins
- Providing a **model** to configure, run and gather results from plugins
- Letting you **customise** the way to visualise results (e.g. sort them according to your needs).

DrTests

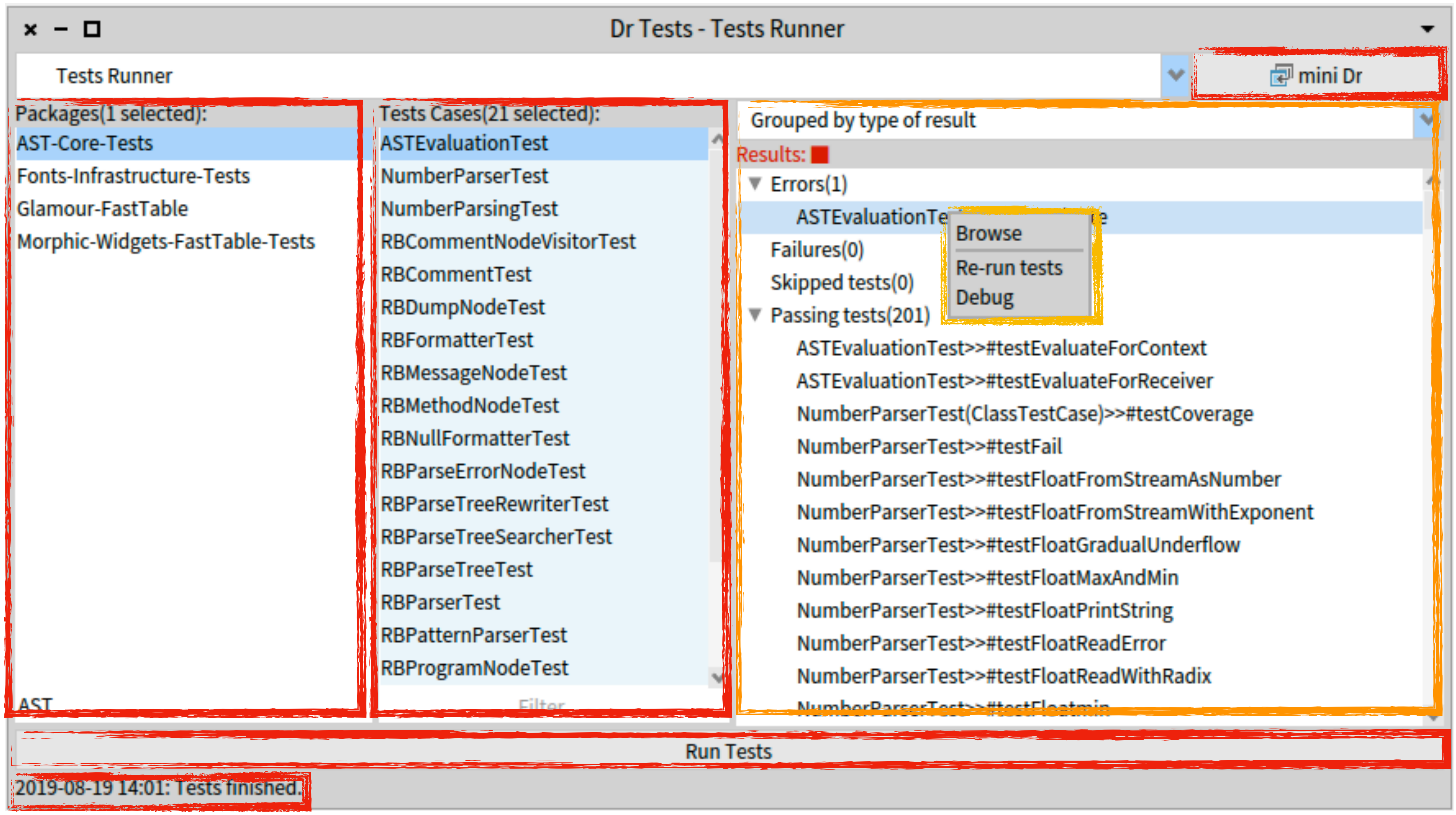
An architecture to build tools around tests

DrTests



```
Dr Tests - Tests Runner
Tests Runner                                   mini Dr
Packages(1 selected):          Tests Cases(21 selected):     Grouped by type of result
AST-Core-Tests                 ASTEvaluationTest             Results: 🟥
Fonts-Infrastructure-Tests     NumberParserTest              ▼ Errors(1)
Glamour-FastTable              NumberParsingTest                 ASTEvaluationTe          te
Morphic-Widgets-FastTable-Tests RBCommentNodeVisitorTest       Failures(0)          Browse
                               RBCommentTest                   Skipped tests(0)     Re-run tests
                               RBDumpNodeTest                ▼ Passing tests(201)   Debug
                               RBFormatterTest                   ASTEvaluationTest>>#testEvaluateForContext
                               RBMessageNodeTest                 ASTEvaluationTest>>#testEvaluateForReceiver
                               RBMethodNodeTest                  NumberParserTest(ClassTestCase)>>#testCoverage
                               RBNullFormatterTest               NumberParserTest>>#testFail
                               RBParseErrorNodeTest              NumberParserTest>>#testFloatFromStreamAsNumber
                               RBParseTreeRewriterTest           NumberParserTest>>#testFloatFromStreamWithExponent
                               RBParseTreeSearcherTest           NumberParserTest>>#testFloatGradualUnderflow
                               RBParseTreeTest                   NumberParserTest>>#testFloatMaxAndMin
                               RBParserTest                      NumberParserTest>>#testFloatPrintString
                               RBPatternParserTest               NumberParserTest>>#testFloatReadError
                               RBProgramNodeTest                 NumberParserTest>>#testFloatReadWithRadix
AST                            Filter...                         NumberParserTest>>#testFloatmin
                                   Run Tests
2019-08-19 14:01: Tests finished.
```

Who has the control?



Dr Tests - Tests Runner

Tests Runner

mini Dr

Packages(1 selected):

- AST-Core-Tests
- Fonts-Infrastructure-Tests
- Glamour-FastTable
- Morphic-Widgets-FastTable-Tests

AST

Tests Cases(21 selected):

- ASTEvaluationTest
- NumberParserTest
- NumberParsingTest
- RBCommentNodeVisitorTest
- RBCommentTest
- RBDumpNodeTest
- RBFormatterTest
- RBMessageNodeTest
- RBMethodNodeTest
- RBNullFormatterTest
- RBParseErrorNodeTest
- RBParseTreeRewriterTest
- RBParseTreeSearcherTest
- RBParseTreeTest
- RBParserTest
- RBPatternParserTest
- RBProgramNodeTest

Filter

Grouped by type of result

Results:

- Errors(1)
  - ASTEvaluationTe
- Failures(0)
- Skipped tests(0)
- Passing tests(201)
  - ASTEvaluationTest>>#testEvaluateForContext
  - ASTEvaluationTest>>#testEvaluateForReceiver
  - NumberParserTest(ClassTestCase)>>#testCoverage
  - NumberParserTest>>#testFail
  - NumberParserTest>>#testFloatFromStreamAsNumber
  - NumberParserTest>>#testFloatFromStreamWithExponent
  - NumberParserTest>>#testFloatGradualUnderflow
  - NumberParserTest>>#testFloatMaxAndMin
  - NumberParserTest>>#testFloatPrintString
  - NumberParserTest>>#testFloatReadError
  - NumberParserTest>>#testFloatReadWithRadix
  - NumberParserTest>>#testFloatmin

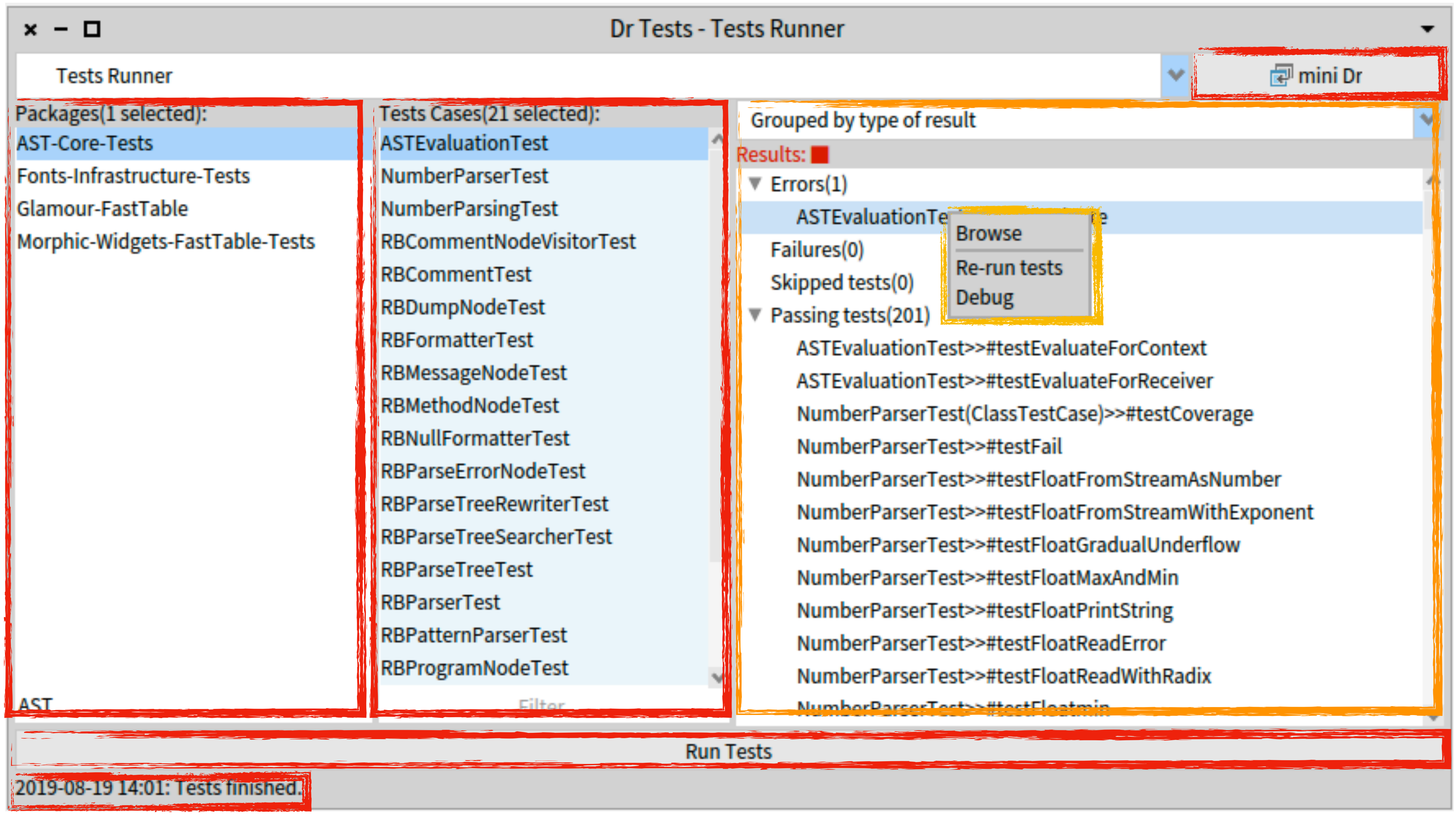
Run Tests

2019-08-19 14:01: Tests finished.

🟥 **Managed by plugin**

🟨 **Managed by result**

Who has the control?



Dr Tests - Tests Runner

Tests Runner

mini Dr

Packages(1 selected):

- AST-Core-Tests
- Fonts-Infrastructure-Tests
- Glamour-FastTable
- Morphic-Widgets-FastTable-Tests

AST

Tests Cases(21 selected):

- ASTEvaluationTest
- NumberParserTest
- NumberParsingTest
- RBCommentNodeVisitorTest
- RBCommentTest
- RBDumpNodeTest
- RBFormatterTest
- RBMessageNodeTest
- RBMethodNodeTest
- RBNullFormatterTest
- RBParseErrorNodeTest
- RBParseTreeRewriterTest
- RBParseTreeSearcherTest
- RBParseTreeTest
- RBParserTest
- RBPatternParserTest
- RBProgramNodeTest

Filter

Grouped by type of result

Results:

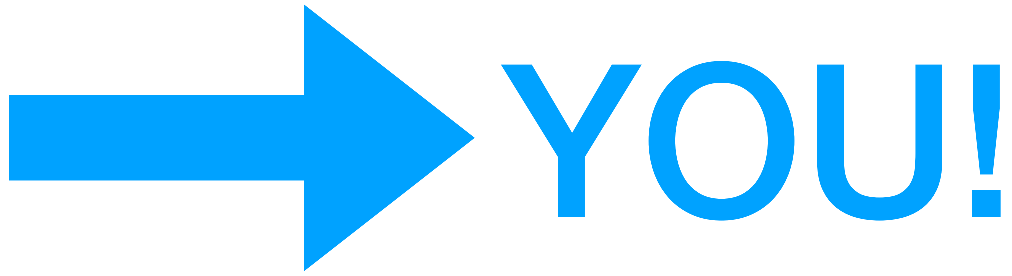
- Errors(1)
  - ASTEvaluationTe
- Failures(0)
- Skipped tests(0)
- Passing tests(201)
  - ASTEvaluationTest>>#testEvaluateForContext
  - ASTEvaluationTest>>#testEvaluateForReceiver
  - NumberParserTest(ClassTestCase)>>#testCoverage
  - NumberParserTest>>#testFail
  - NumberParserTest>>#testFloatFromStreamAsNumber
  - NumberParserTest>>#testFloatFromStreamWithExponent
  - NumberParserTest>>#testFloatGradualUnderflow
  - NumberParserTest>>#testFloatMaxAndMin
  - NumberParserTest>>#testFloatPrintString
  - NumberParserTest>>#testFloatReadError
  - NumberParserTest>>#testFloatReadWithRadix
  - NumberParserTest>>#testFloatmin

Run Tests

2019-08-19 14:01: Tests finished.

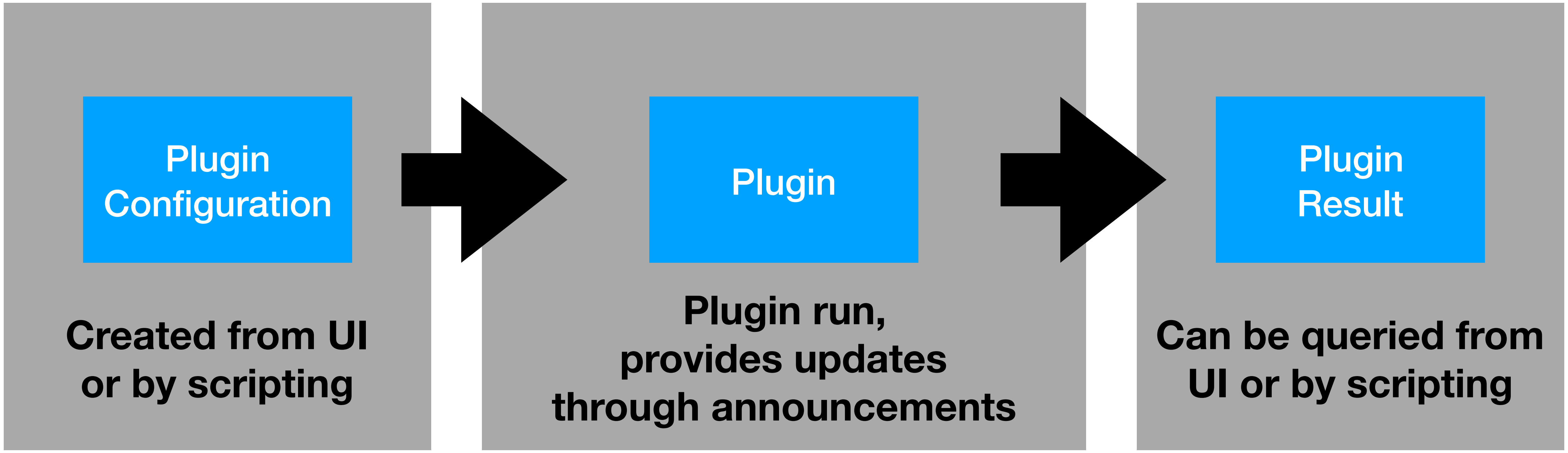
🟥 **Managed by plugin**

🟨 **Managed by result**



Demo

DrTests’ model

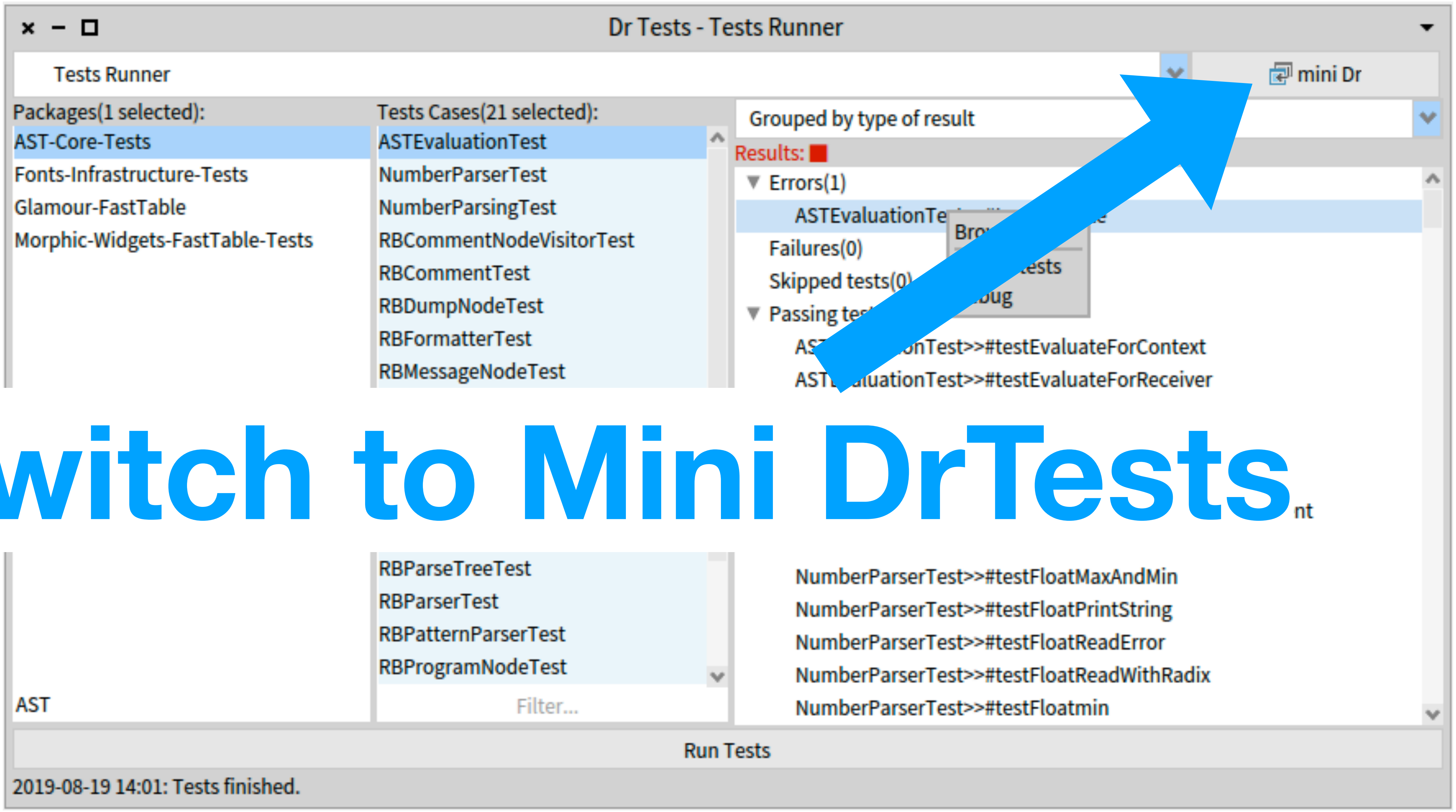

Plugin Configuration
Created from UI or by scripting
Plugin
Plugin run, provides updates through announcements
Plugin Result
Can be queried from UI or by scripting

Consequence of DrTests' model

Plugins can be exposed via different UIs easily

Mini DrTests

DrTests



Dr Tests - Tests Runner

Tests Runner | mini Dr

Packages(1 selected):
AST-Core-Tests
Fonts-Infrastructure-Tests
Glamour-FastTable
Morphic-Widgets-FastTable-Tests
AST

Tests Cases(21 selected):
ASTEvaluationTest
NumberParserTest
NumberParsingTest
RBCommentNodeVisitorTest
RBCommentTest
RBDumpNodeTest
RBFormatterTest
RBMessageNodeTest
RBParseTreeTest
RBParserTest
RBPatternParserTest
RBProgramNodeTest
Filter...

Grouped by type of result

Results: 🟥

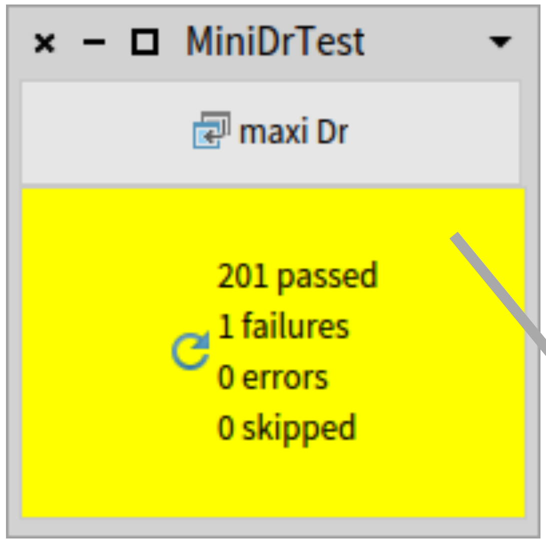
▼ Errors(1)
ASTEvaluationTe
Failures(0)
Skipped tests(0)
▼ Passing te
AS nTest>>#testEvaluateForContext
AST aluationTest>>#testEvaluateForReceiver
NumberParserTest>>#testFloatMaxAndMin
NumberParserTest>>#testFloatPrintString
NumberParserTest>>#testFloatReadError
NumberParserTest>>#testFloatReadWithRadix
NumberParserTest>>#testFloatmin

Run Tests

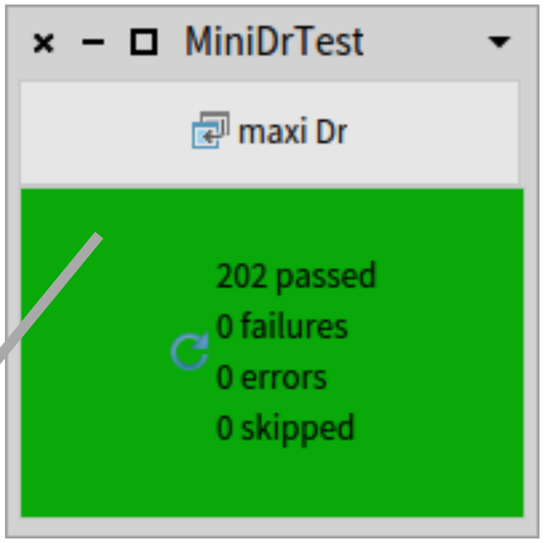
2019-08-19 14:01: Tests finished.

Switch to Mini DrTests

Mini DrTests

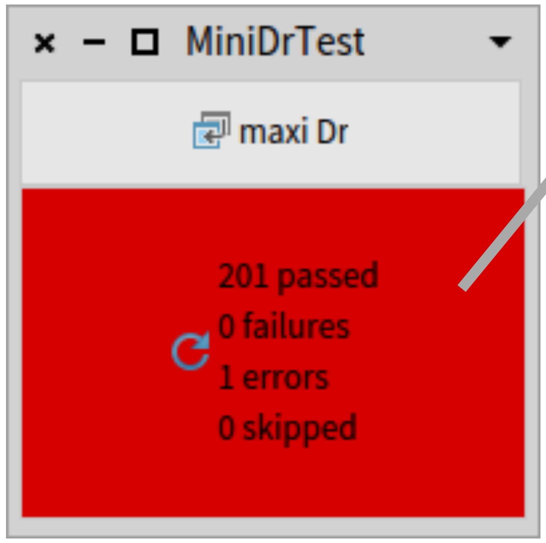


```
201 passed
1 failures
0 errors
0 skipped
```

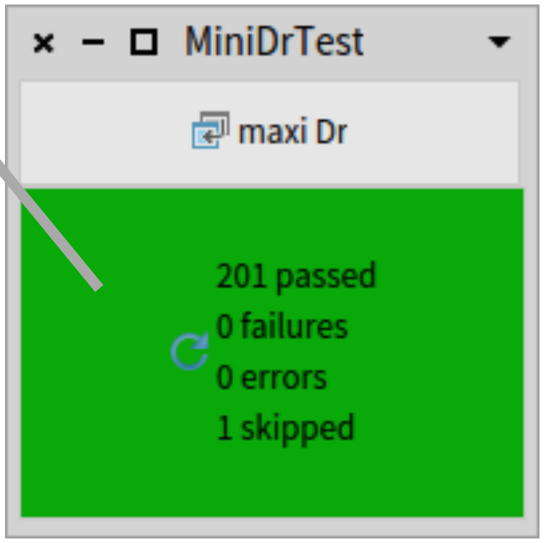


```
202 passed
0 failures
0 errors
0 skipped
```

Quick visual feedback



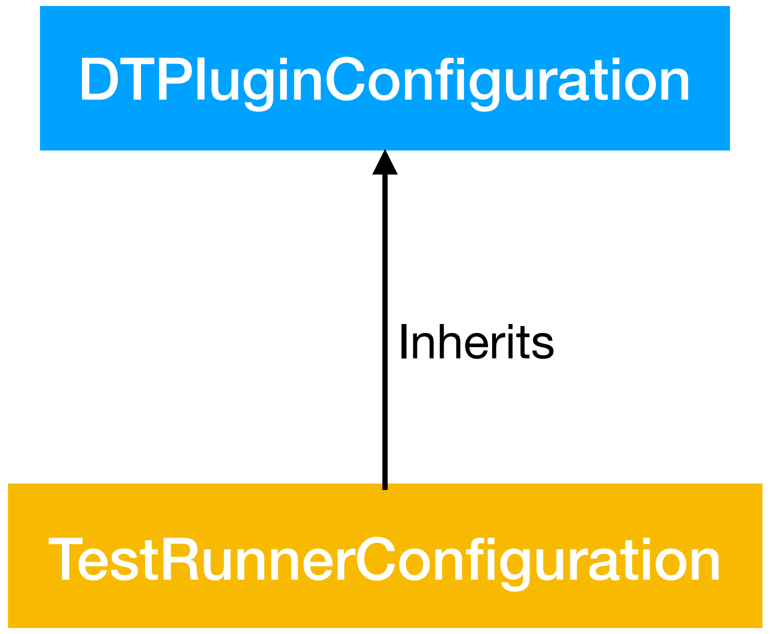
```
201 passed
0 failures
1 errors
0 skipped
```



```
201 passed
0 failures
0 errors
1 skipped
```

DrTest API

In a nutshell

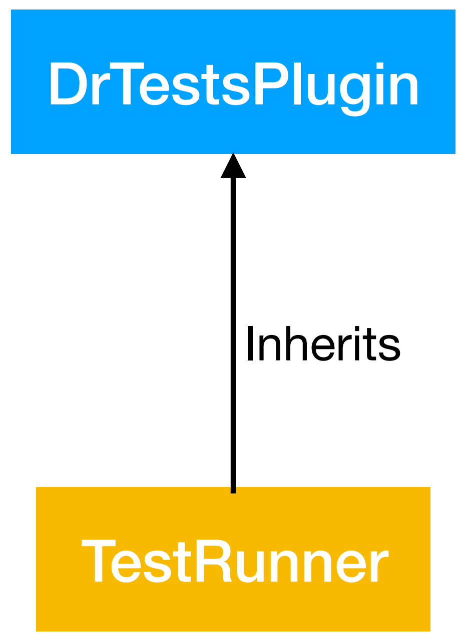


- Stores data required for the plugin to run in instance variables

```
DTPluginConfiguration subclass: #DTTestsRunnerConfiguration
    slots: { #tests }
    classVariables: {  }
    package: 'DrTests-TestsRunner'
```

DrTest API

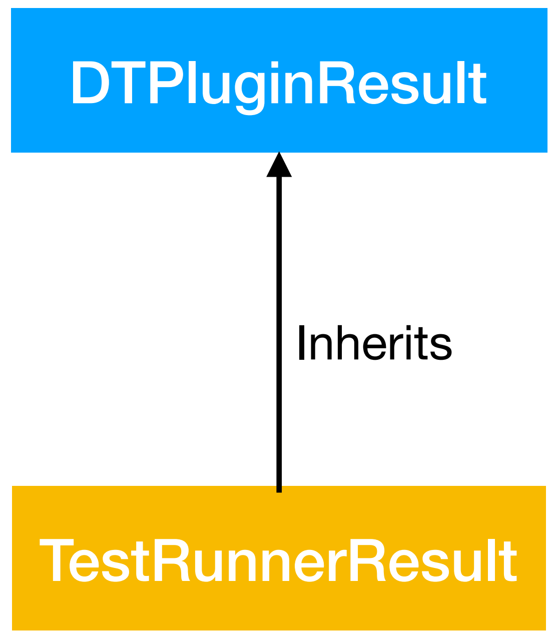
In a nutshell



- Set UI labels for lists, tree and buttons.
- Describe how to fill lists to let user create a configuration
- Create configuration by reading UI’s state in #buildConfigurationFrom:
- Define how to run the plugin in #runForConfiguration:
- Define if can be minified or not via #allowMiniDrTests

DrTest API

In a nutshell



- Define how to build UI trees via pragmas
- Define actions available for nodes via commands

DrTest API

In a nutshell

➡Define how to build result trees via pragmas

```
buildGroupedByTypeTree
    <dTTestRunnerResultTreeNamed: 'Grouped by type of result'>
    ^ DTTreeNode new
        subResults:
            {DTTreeNode new
                name: DTTestResultType error pluralName;
                subResults: (self buildLeavesFrom: self testsResult errors type:
DTTestResultType error);
                yourself.
            …
            };
        yourself
```

DrTest API

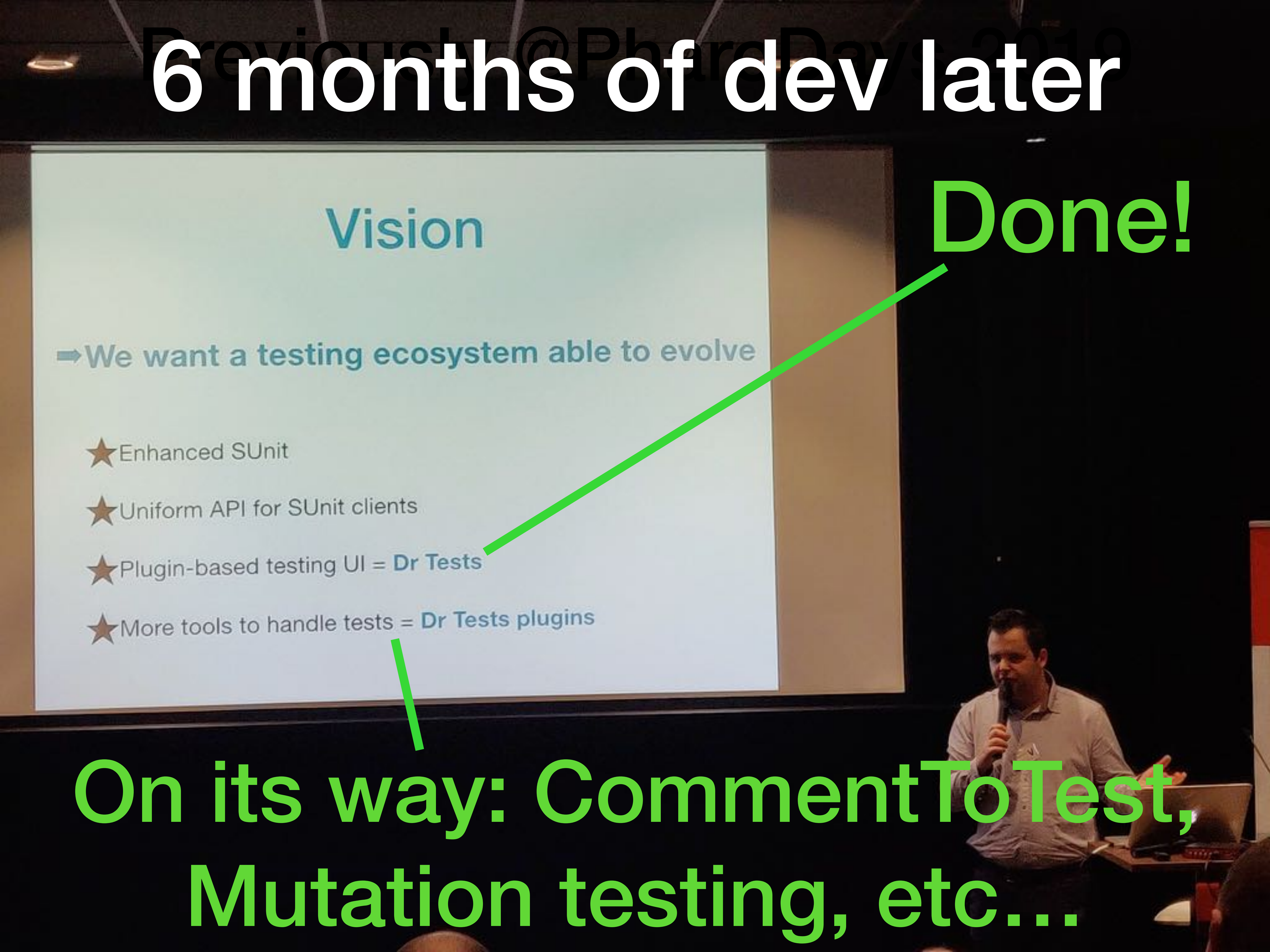
In a nutshell

➡Define actions available for nodes via commands

```
buildContextualMenuGroupWith: presenterIntance
     ^ (CmCommandGroup named: 'TestRunnerResult context menu')
asSpecGroup
          basicDescription: 'Commands related to results.';
          register: (DTRerunCommand forSpec context: presenterIntance)
                       beHiddenWhenCantBeRun;
          register: (DTDebugTestCommand forSpec context: presenterIntance)
                       beHiddenWhenCantBeRun;
          beDisplayedAsGroup;
          yourself
```

Previously @PharoDays 2019

6 months of dev later



Vision

➡We want a testing ecosystem able to evolve

- ★Enhanced SUnit
- ★Uniform API for SUnit clients
- ★Plugin-based testing UI = **Dr Tests**
- ★More tools to handle tests = **Dr Tests plugins**

Done!

On its way: CommentToTest, Mutation testing, etc…

Next steps

- Calypso integration
- Enhance SUnit
- Unify SUnit’s API tools are exposed to

Damien on github

Conclusion

- DrTests will be **part of Pharo 8**, SUnit runner will be deprecated
- The new infrastructure allow people to **plug their analyses**
- Tests are super-valuable, DrTests will help to **extract the gold out of them**

Dr Tests opens a lot of possibilities of future tools around tests!

@juldelplanque

juliendelplanque/DrTests

juliendelplanque