

David Kuster
dave@talldave.net
@ofTallDaveFame

REPL Driven Development

With the Grails Console Plugin

Or, Why You Should Be Writing Code In The Browser

About Me

- Grails dev ~4 years
- Java/JEE dev ~10 years prior
- Bootstrapped startups to Fortune 500
- Developer of Grails Command Center
- “You're really good at writing your codes.”
- my wife

Agenda

- What the heck's a REPL anyway?
- Console Plugin Demo, Tips & Tricks
- Existing Development Processes
- REPL Driven Development
- The Future

REPL what?

read-eval-print loop

```
(loop (print (eval (read))))
```

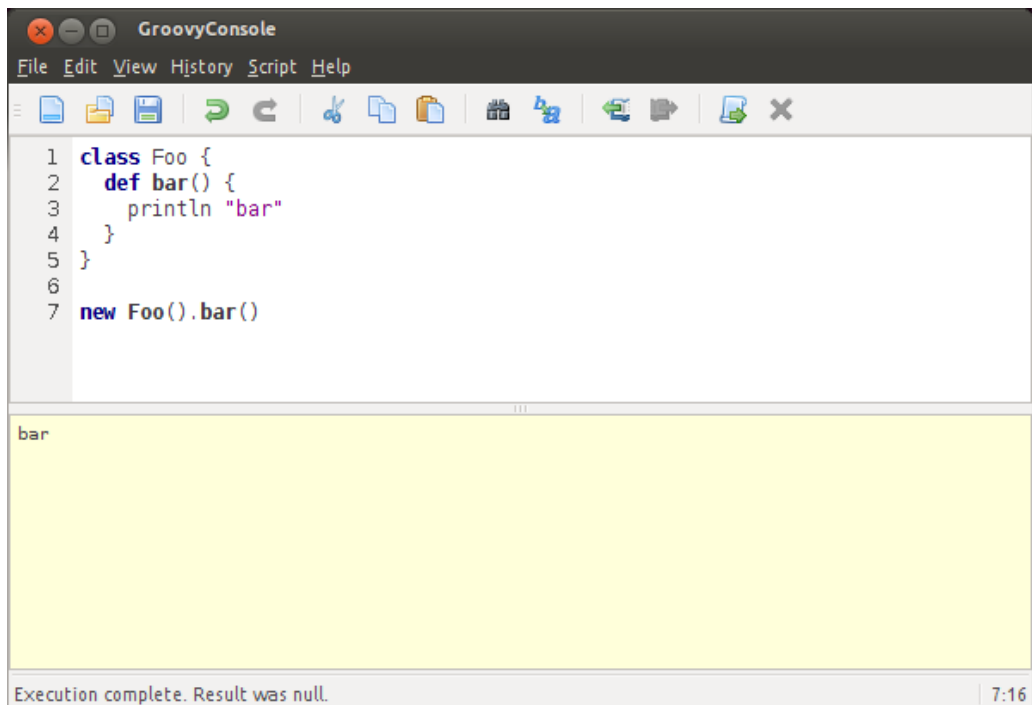
Groovy Ecosystem

- groovysh
- groovyConsole
- grails console
- Grails Console Plugin

groovysh

```
david@david-ThinkPad-T430s: ~  
david@david-ThinkPad-T430s:~$ opt/groovy-2.3.4/bin/groovysh  
Groovy Shell (2.3.4, JVM: 1.7.0_51)  
Type ':help' or ':h' for help.  
-----  
groovy:000> "This is a sentence" =~ "is"  
==> java.util.regex.Matcher[pattern=is region=0,18 lastmatch=]  
groovy:000> ("This is a sentence" =~ "is") as Boolean  
==> true  
groovy:000> ("This is a sentence" =~ "isn't") as Boolean  
==> false  
groovy:000>  
groovy:000>  
groovy:000> ['1','2','3'].remove('2')  
==> true  
groovy:000> ['1','2','3'].remove(2)  
==> 3  
groovy:000> 
```

groovyConsole/grails console

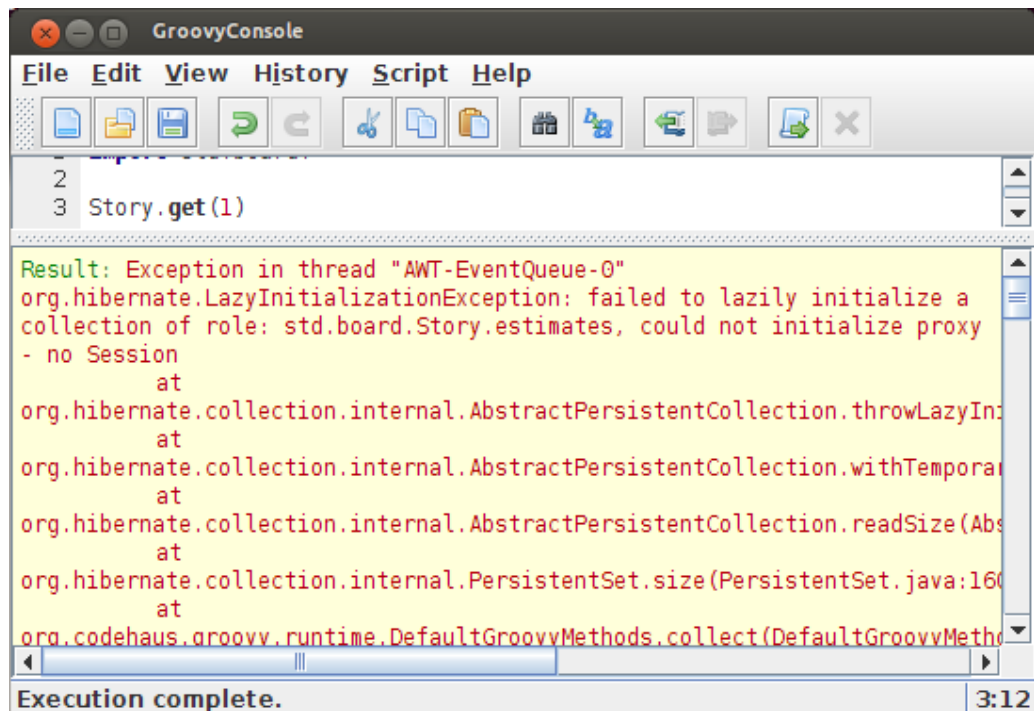


The screenshot shows a window titled "GroovyConsole" with a menu bar (File, Edit, View, History, Script, Help) and a toolbar. The main text area contains the following Groovy code:

```
1 class Foo {  
2     def bar() {  
3         println "bar"  
4     }  
5 }  
6  
7 new Foo().bar()
```

Below the code area, the output "bar" is displayed on a yellow background. At the bottom of the window, a status bar shows "Execution complete. Result was null." and the time "7:16".

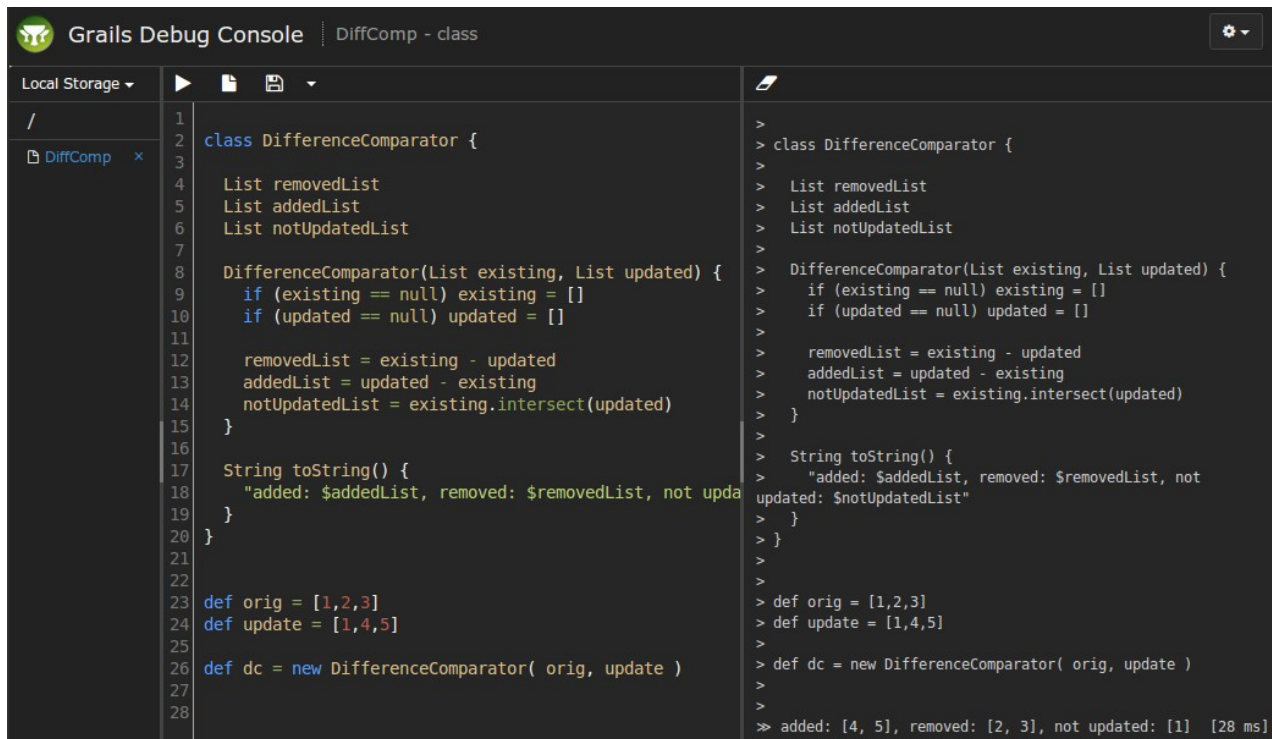
grails console



The screenshot shows a window titled "GroovyConsole" with a menu bar (File, Edit, View, History, Script, Help) and a toolbar with icons for file operations and execution. The console area displays a script with two lines: line 2 is empty, and line 3 contains the code `Story.get(1)`. Below the script, the output shows a red exception message: "Result: Exception in thread 'AWT-EventQueue-0'" followed by a stack trace for `org.hibernate.LazyInitializationException`. The stack trace includes the following frames: `failed to lazily initialize a collection of role: std.board.Story.estimateds, could not initialize proxy - no Session`, `at org.hibernate.collection.internal.AbstractPersistentCollection.throwLazyInit...`, `at org.hibernate.collection.internal.AbstractPersistentCollection.withTempor...`, `at org.hibernate.collection.internal.AbstractPersistentCollection.readSize(Abs...`, `at org.hibernate.collection.internal.PersistentSet.size(PersistentSet.java:160)`, and `at org.codehaus.groovy.runtime.DefaultGroovyMethods.collect(DefaultGroovyMeth...`. The status bar at the bottom indicates "Execution complete." and the time "3:12".

```
2  
3 Story.get(1)  
  
Result: Exception in thread "AWT-EventQueue-0"  
org.hibernate.LazyInitializationException: failed to lazily initialize a  
collection of role: std.board.Story.estimateds, could not initialize proxy  
- no Session  
    at  
    org.hibernate.collection.internal.AbstractPersistentCollection.throwLazyInit  
    at  
    org.hibernate.collection.internal.AbstractPersistentCollection.withTempor  
    at  
    org.hibernate.collection.internal.AbstractPersistentCollection.readSize(Abs  
    at  
    org.hibernate.collection.internal.PersistentSet.size(PersistentSet.java:160  
    at  
    org.codehaus.groovy.runtime.DefaultGroovyMethods.collect(DefaultGroovyMeth  
  
Execution complete. 3:12
```


Console Plugin



The screenshot displays the Grails Debug Console interface. The title bar shows the Grails logo, 'Grails Debug Console', and the active class 'DiffComp - class'. On the left, a 'Local Storage' pane shows the file structure with 'DiffComp' selected. The main editor area is split into two panes. The left pane shows the source code of the `DiffComp` class, with line numbers 1 through 28. The right pane shows the same code with green annotations indicating the execution flow. The code defines a `DifferenceComparator` class with attributes `removedList`, `addedList`, and `notUpdatedList`, a constructor, and a `toString()` method. It also includes test code defining `orig` and `update` lists, creating a `DiffComp` instance, and printing the result. The output at the bottom shows the result of the `toString()` method call: `>> added: [4, 5], removed: [2, 3], not updated: [1] [28 ms]`.

```
1 class DifferenceComparator {
2
3
4     List removedList
5     List addedList
6     List notUpdatedList
7
8     DifferenceComparator(List existing, List updated) {
9         if (existing == null) existing = []
10        if (updated == null) updated = []
11
12        removedList = existing - updated
13        addedList = updated - existing
14        notUpdatedList = existing.intersect(updated)
15    }
16
17    String toString() {
18        "added: $addedList, removed: $removedList, not upda
19    }
20 }
21
22
23 def orig = [1,2,3]
24 def update = [1,4,5]
25
26 def dc = new DifferenceComparator( orig, update )
27
28
```

```
> class DifferenceComparator {
>
>     List removedList
>     List addedList
>     List notUpdatedList
>
>     DifferenceComparator(List existing, List updated) {
>         if (existing == null) existing = []
>         if (updated == null) updated = []
>
>         removedList = existing - updated
>         addedList = updated - existing
>         notUpdatedList = existing.intersect(updated)
>     }
>
>     String toString() {
>         "added: $addedList, removed: $removedList, not
>         updated: $notUpdatedList"
>     }
> }
>
>
> def orig = [1,2,3]
> def update = [1,4,5]
>
> def dc = new DifferenceComparator( orig, update )
>
>
>> added: [4, 5], removed: [2, 3], not updated: [1] [28 ms]
```

Shortcuts

Shortcuts

Ctrl-Enter / Cmd-Enter

Execute

Ctrl-S / Cmd-S

Save

Esc

Clear Output

Implicit Variables

Implicit variables

`ctx`

the Spring application context

`grailsApplication`

the Grails application

`config`

the Grails configuration

`request`

the HTTP request

`session`

the HTTP session

Environment Availability

- Recommend only including in dev/test environments
- Can include in prod but...
 - Definitely secure the /console URL
 - Be careful
 - Domain.list()*.delete()
 - Oops!

```
// BuildConfig.groovy
import grails.util.Environment

plugins {
    ...
    if (Environment.current != Environment.PRODUCTION) {
        compile ":console:1.4.4"
    }
}
```

Agenda

- What the heck's a REPL anyway?
- Console Plugin Demo, Tips & Tricks
- Existing Development Processes
- REPL Driven Development
- The Future

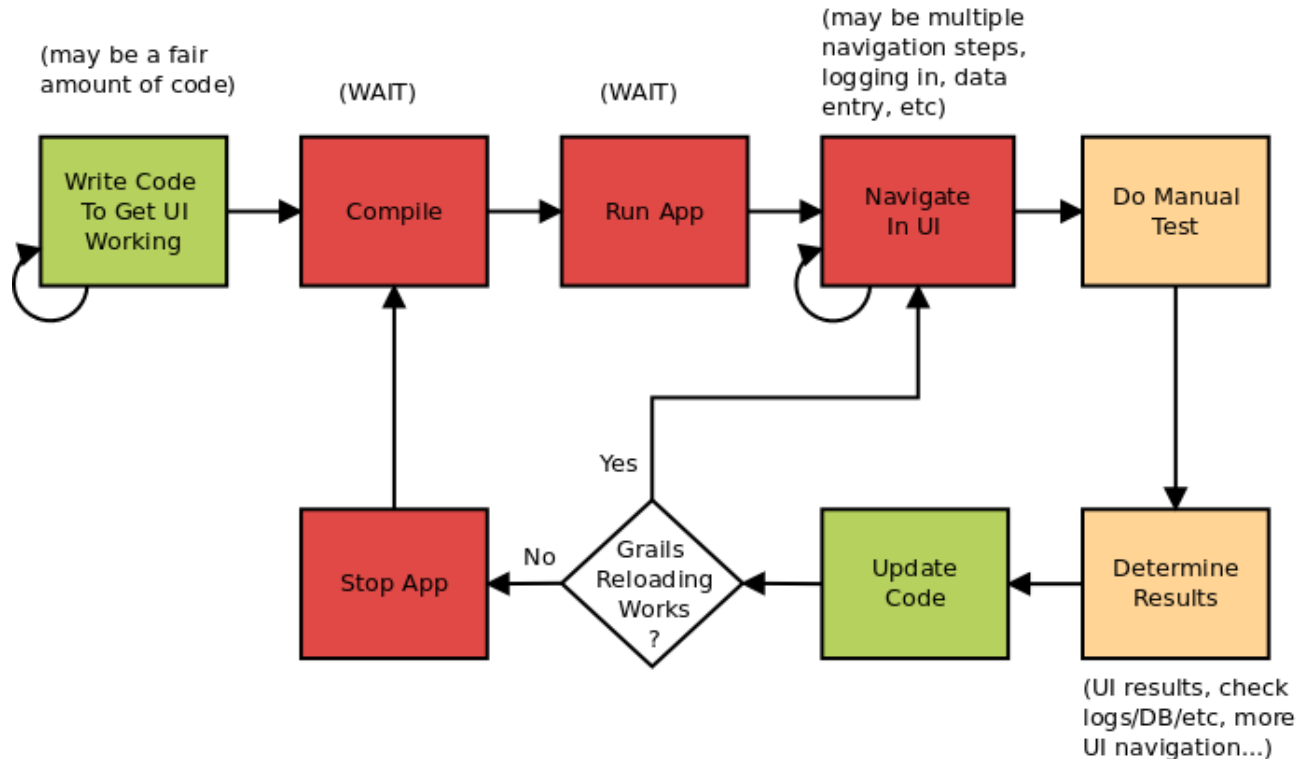
Agenda

- What the heck's a REPL anyway?
- Console Plugin Demo, Tips & Tricks
- Existing Development Processes
- REPL Driven Development
- The Future

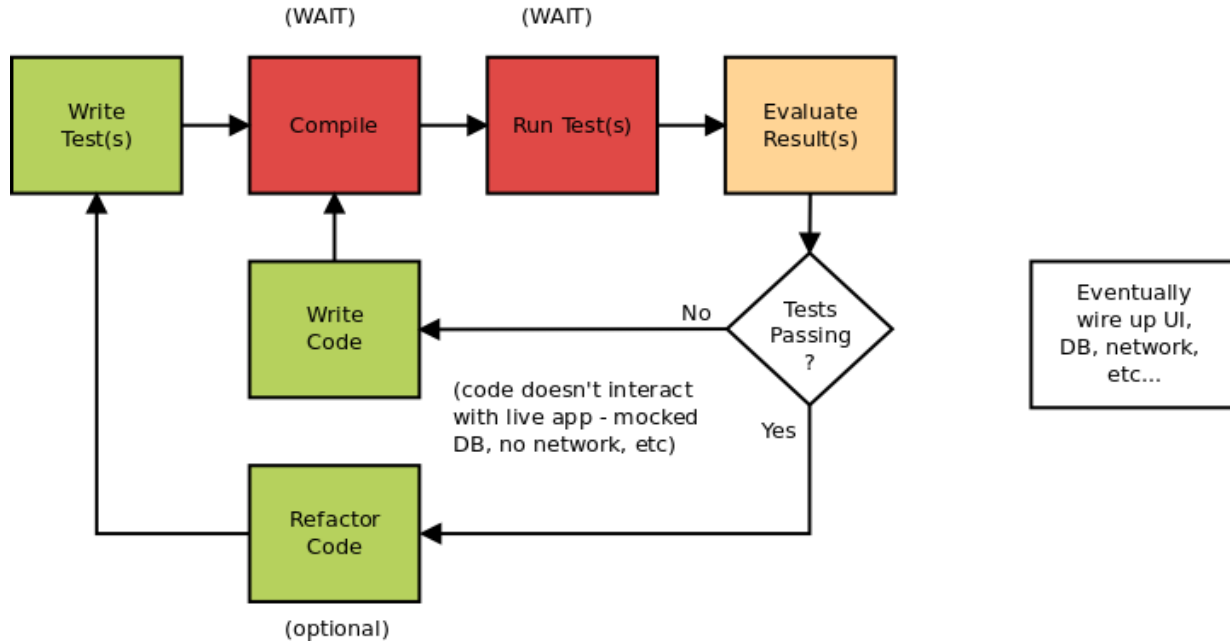
A Sampling of Dev Processes

- UI / Scaffolding
- TDD
- BDD
- FDD

UI / Scaffolding Driven

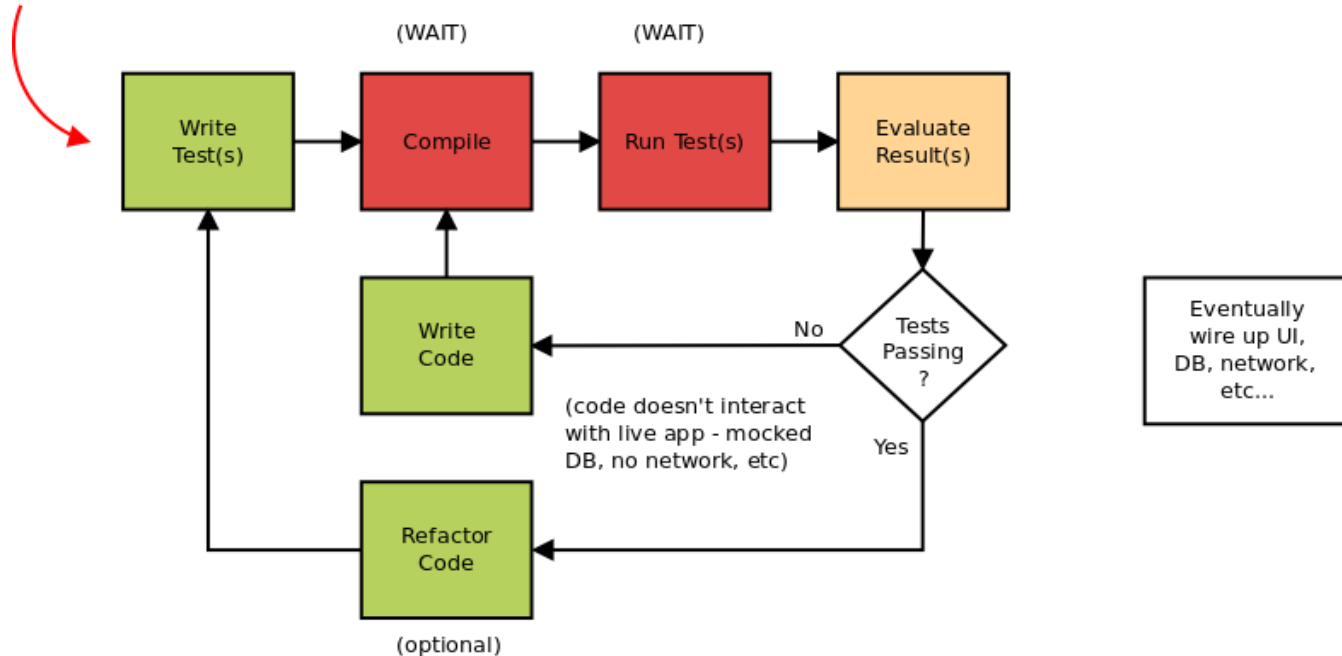


Test Driven



Behavior Driven

SHOULD



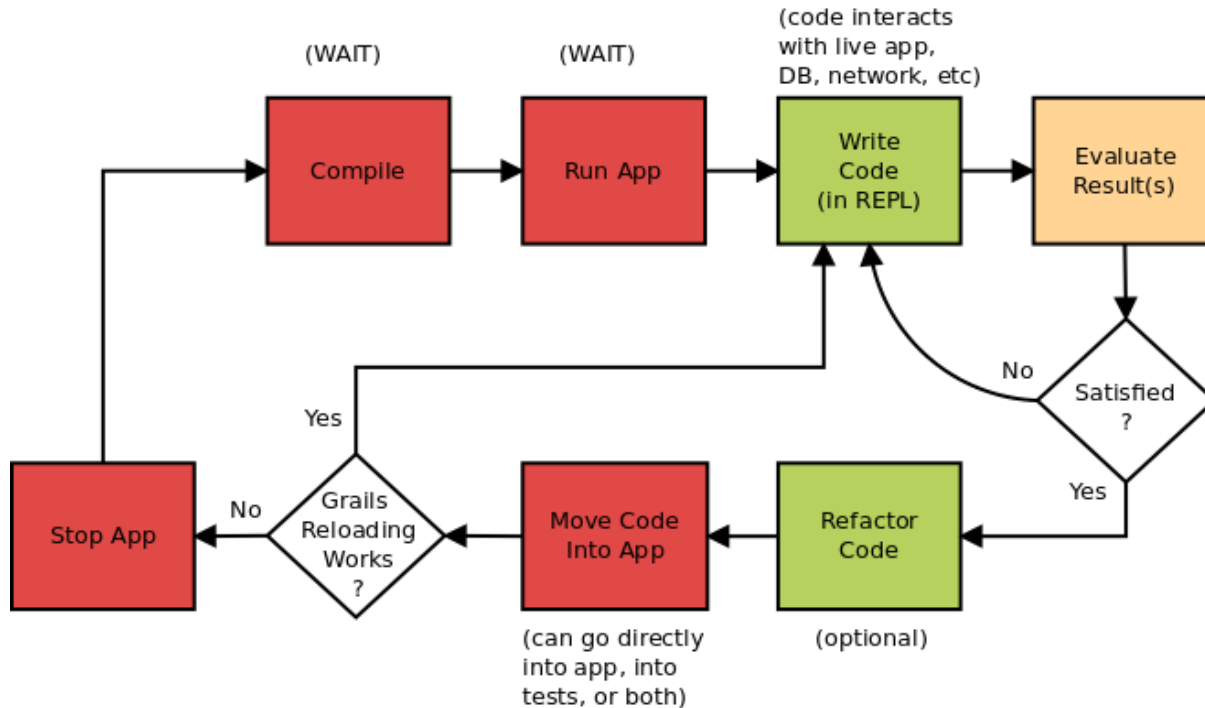
Faith Driven



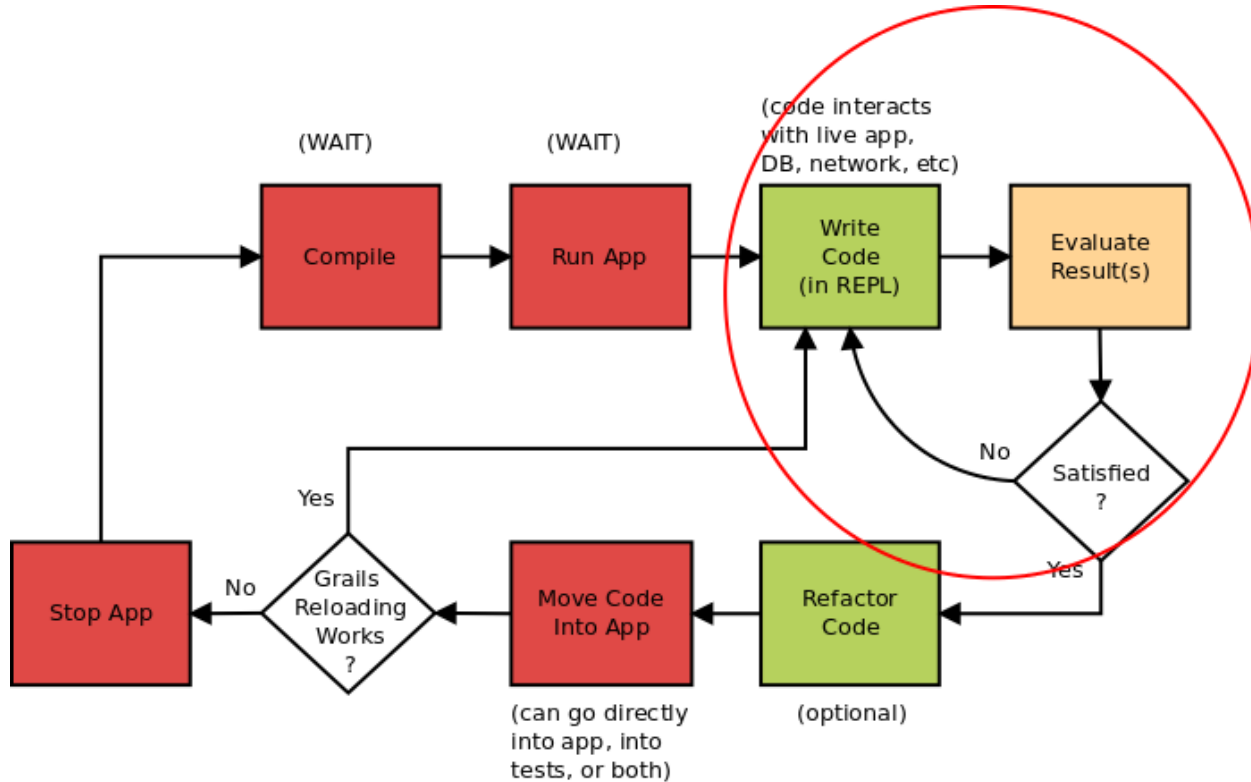
Agenda

- What the heck's a REPL anyway?
- Console Plugin Demo, Tips & Tricks
- Existing Development Processes
- **REPL Driven Development**
- The Future

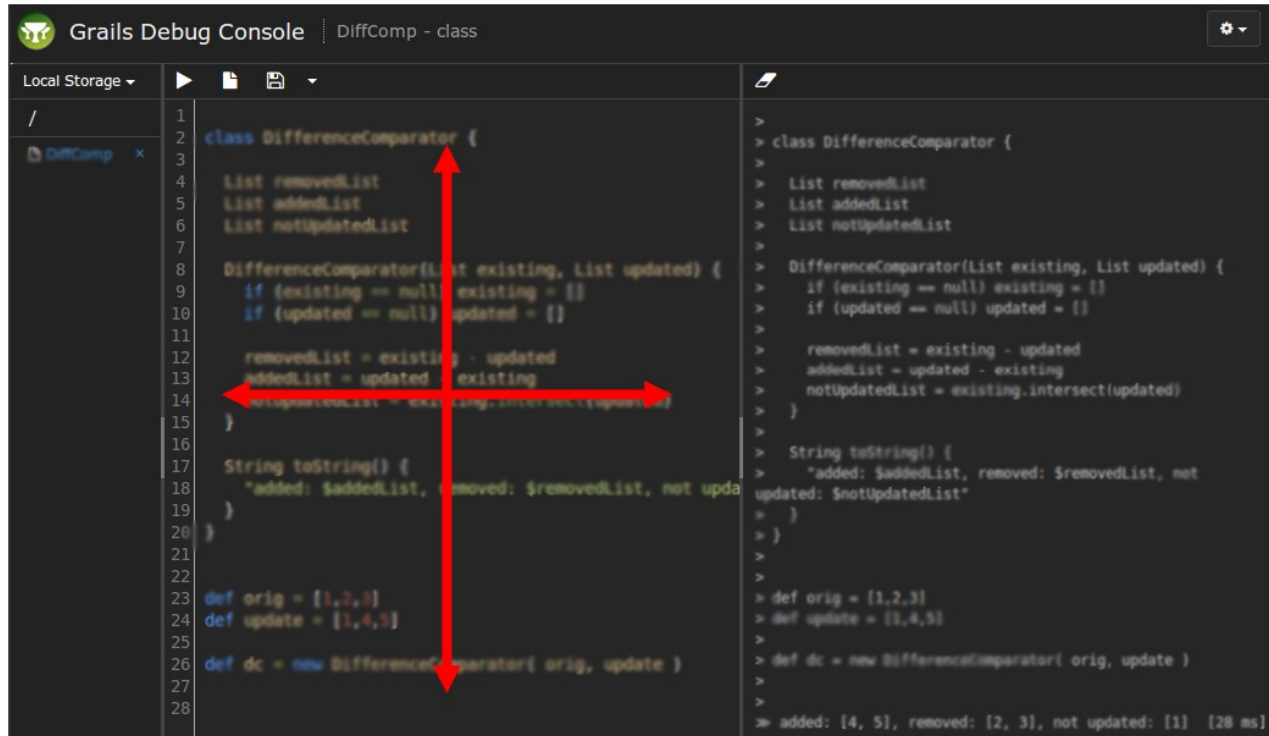
REPL Driven



Where We Spend Our Time



Constraint Creates Innovation



The image shows a Grails Debug Console window with the title "Grails Debug Console | DiffComp - class". The interface is split into two main panes. The left pane shows the source code for a class named "DifferenceComparator". The right pane shows the same class code as it appears in the REPL, with some additional context and output.

Local Storage ▾

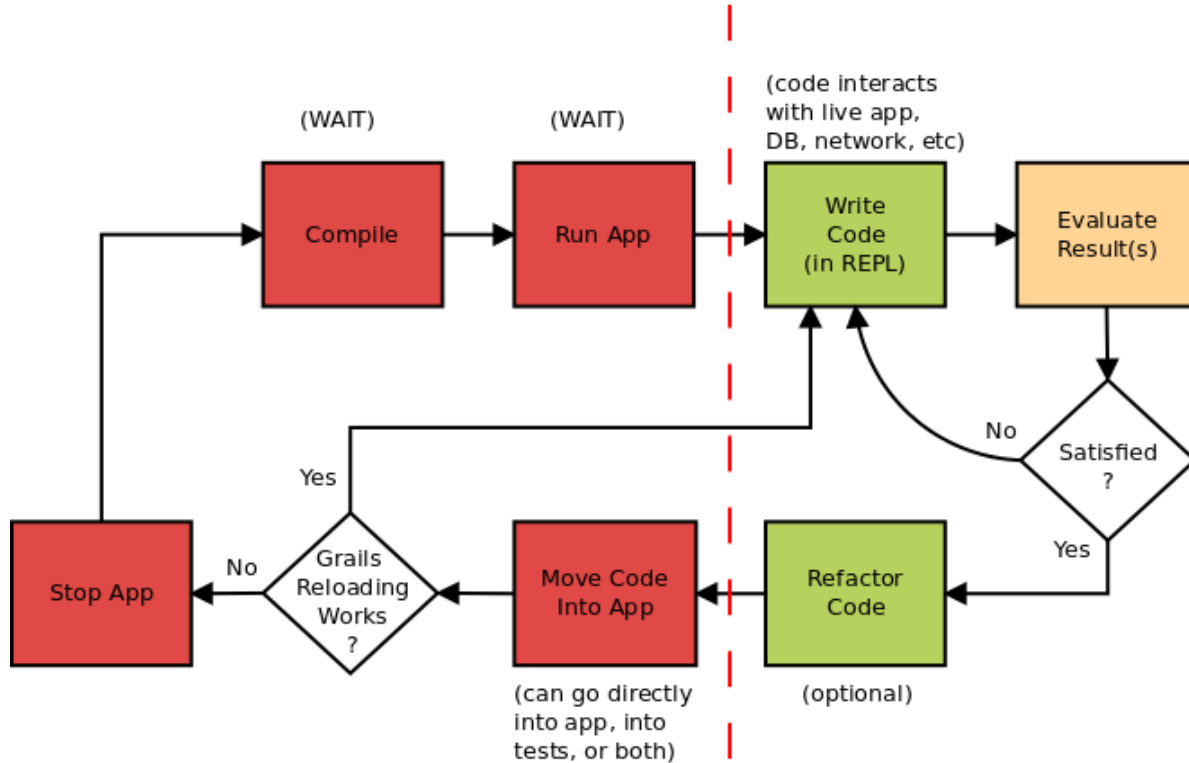
/

DiffComp x

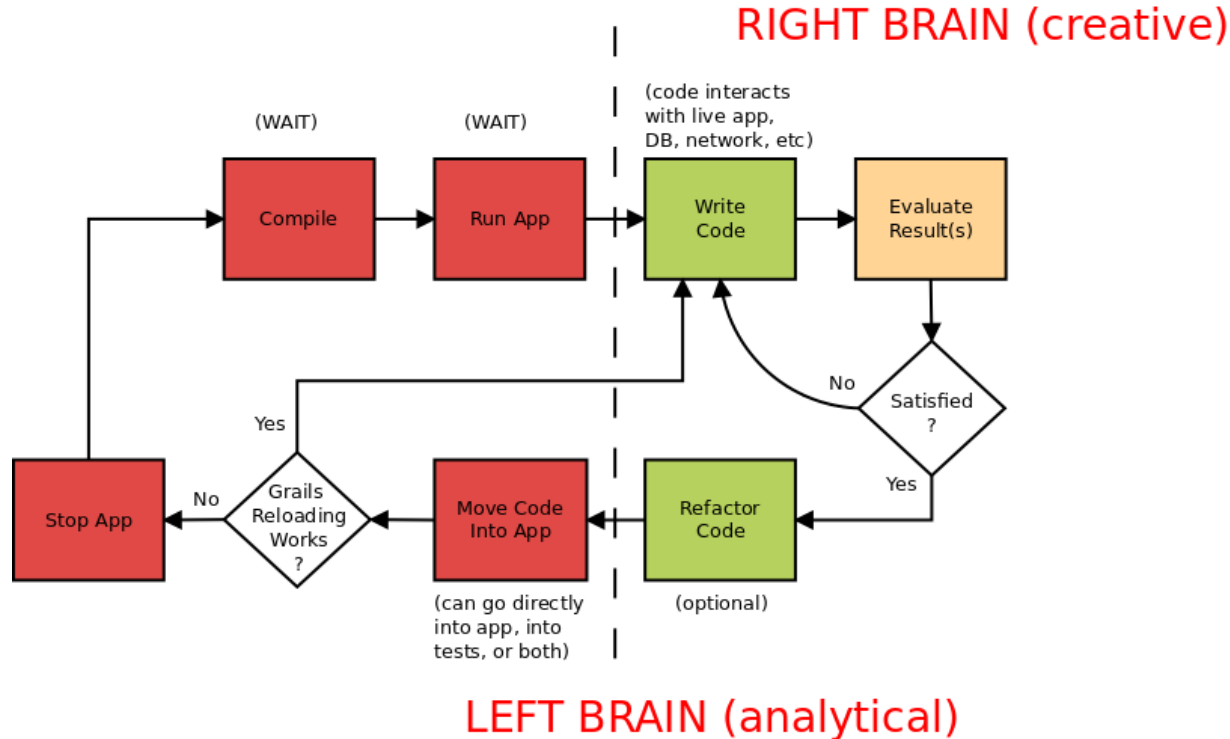
```
1 class DifferenceComparator {
2
3
4   List removedList
5   List addedList
6   List notUpdatedList
7
8   DifferenceComparator(List existing, List updated) {
9     if (existing == null) existing = []
10    if (updated == null) updated = []
11
12    removedList = existing - updated
13    addedList = updated - existing
14    notUpdatedList = existing.intersect(updated)
15  }
16
17  String toString() {
18    "added: $addedList, removed: $removedList, not updated: $notUpdatedList"
19  }
20 }
21
22
23 def orig = [1,2,3]
24 def update = [1,4,5]
25
26 def dc = new DifferenceComparator( orig, update )
27
28
```

> class DifferenceComparator {
> List removedList
> List addedList
> List notUpdatedList
> DifferenceComparator(List existing, List updated) {
> if (existing == null) existing = []
> if (updated == null) updated = []
> removedList = existing - updated
> addedList = updated - existing
> notUpdatedList = existing.intersect(updated)
> }
> String toString() {
> "added: \$addedList, removed: \$removedList, not updated: \$notUpdatedList"
> }
> }
>
> def orig = [1,2,3]
> def update = [1,4,5]
> def dc = new DifferenceComparator(orig, update)
>
> added: [4, 5], removed: [2, 3], not updated: [1] [28 ms]

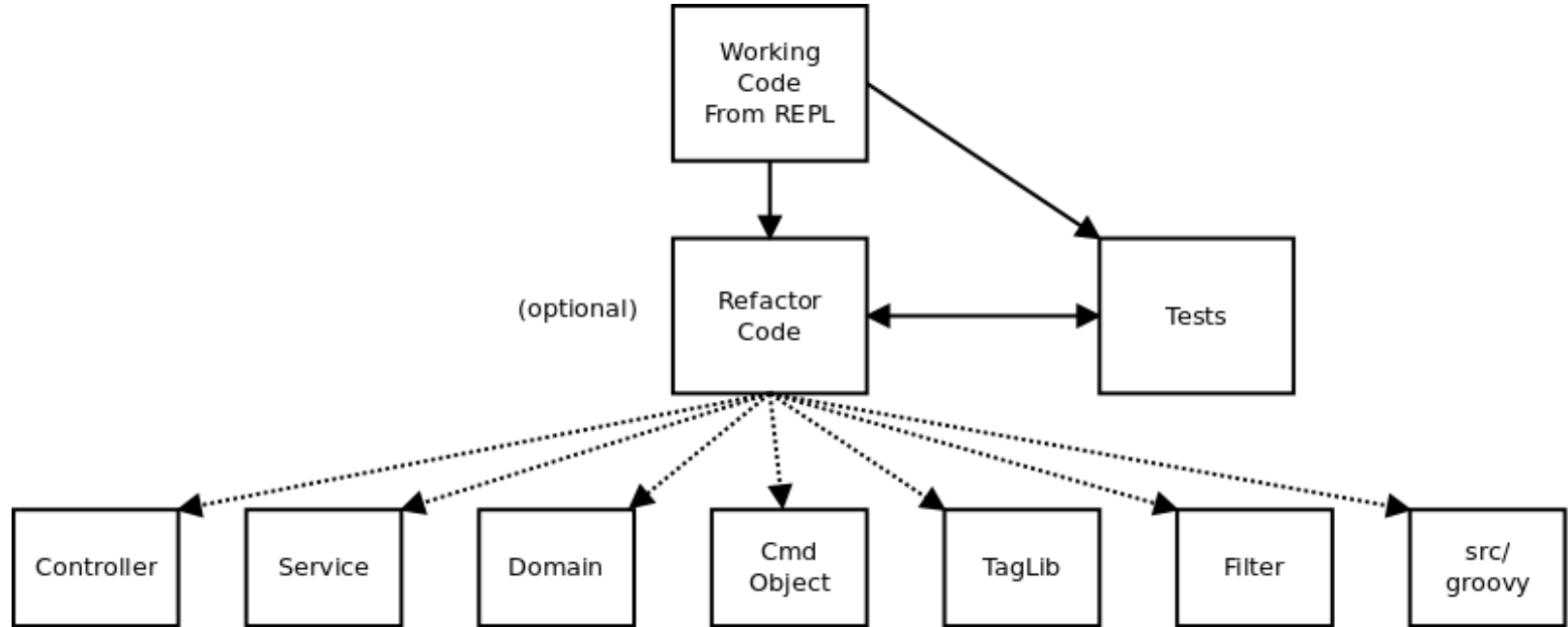
Separation of Code



Right Brain, Left Brain



Integration



Agenda

- What the heck's a REPL anyway?
- Console Plugin Demo, Tips & Tricks
- Existing Development Processes
- REPL Driven Development
- The Future

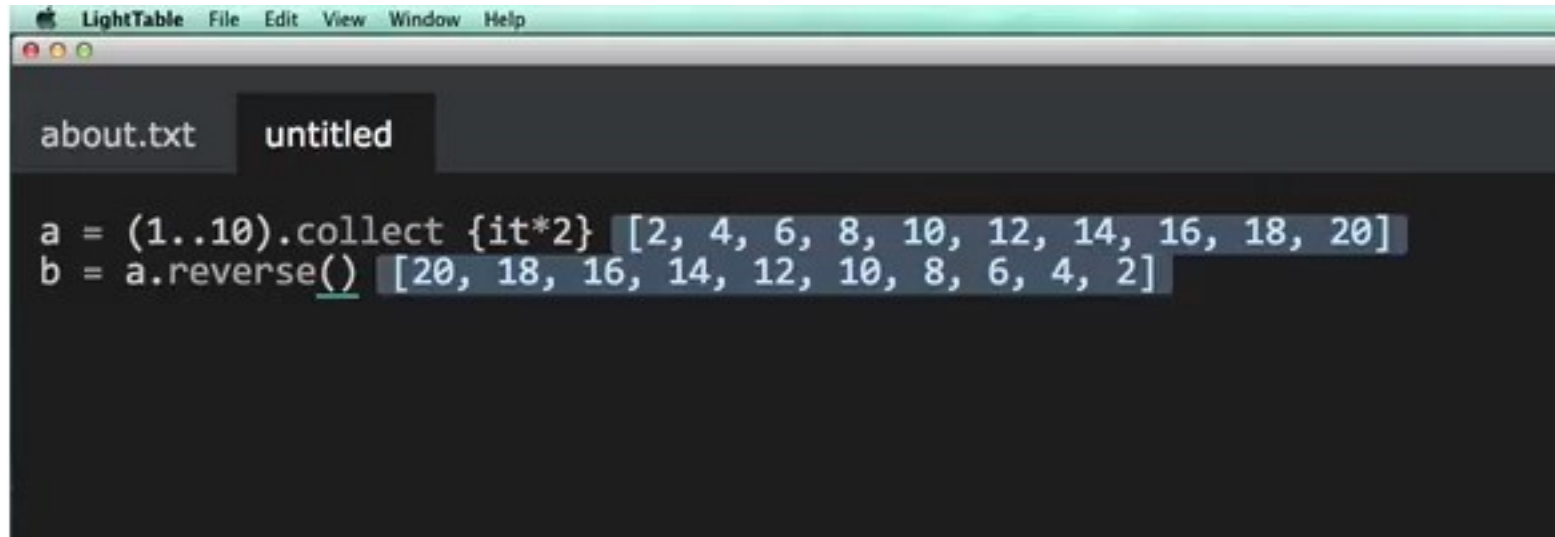
THE FUTURE

- Better Tooling
- REPL Driven Testing
- Documentation

THE FUTURE

“The future is here, it's just not very evenly distributed.”

Light Table – Groovy Plugin



The screenshot shows the LightTable IDE interface. The title bar includes an Apple logo, the text 'LightTable', and a menu bar with 'File', 'Edit', 'View', 'Window', and 'Help'. Below the title bar, there are two tabs: 'about.txt' and 'untitled'. The 'untitled' tab is active, displaying Groovy code. The code consists of two lines: `a = (1..10).collect {it*2}` followed by the result `[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]`, and `b = a.reverse()` followed by the result `[20, 18, 16, 14, 12, 10, 8, 6, 4, 2]`. The results are displayed in a light blue highlight. The 'reverse()' method in the second line is underlined.

```
a = (1..10).collect {it*2} [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
b = a.reverse() [20, 18, 16, 14, 12, 10, 8, 6, 4, 2]
```

Spock Web Console

Spock Web Console

```
1  import spock.lang.*
2
3  // Hit 'Run Script' below
4  class MyFirstSpec extends Specification {
5      def "let's try this!"() {
6          expect:
7              Math.max(1, 2) == 3
8      }
9  }
10
11
12
13
14
15
16
```



Actions > Run Script New Script Publish Script View Recent Scripts



Subscribe

Result Output Stack Trace

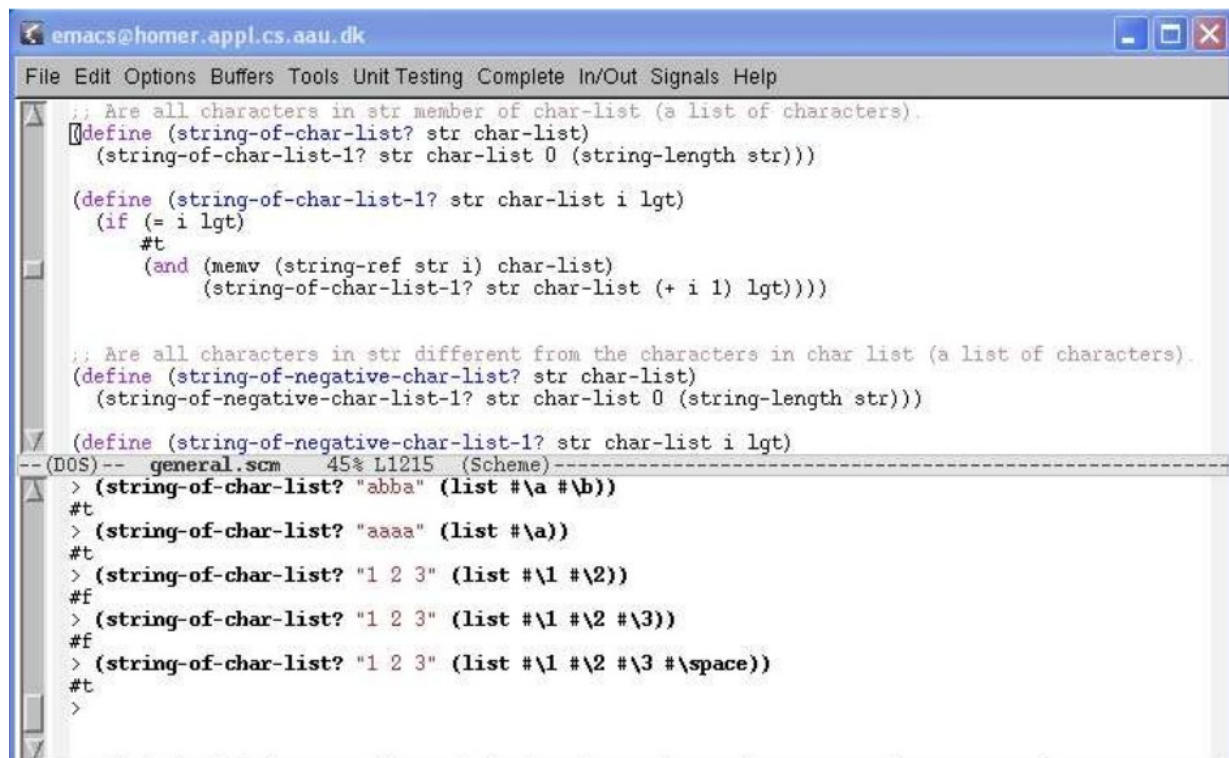
Tips & Tricks

- To run a script, press Alt-Meta-R
- To resize the script area, drag the horizontal rule

About Spock Web Console

- Based on [Groovy Web Console](#), code hosted on [GitHub](#)
- Powered by [Groovy 2.1](#) and [Spock 1.0-SNAPSHOT](#)
- Syntax highlighting provided by [CodeMirror](#) and [SyntaxHighlighter](#)

Scheme Mini-IDE



The screenshot shows an Emacs window titled 'emacs@homer.appl.cs.aau.dk'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'UnitTesting', 'Complete', 'In/Out', 'Signals', and 'Help'. The main text area contains Scheme code for string and character list operations. A horizontal separator line is present, with '(DOS) -- general.scm 45% L1215 (Scheme) -----' below it. The bottom section shows the REPL with several test cases and their results.

```
;; Are all characters in str member of char-list (a list of characters).
(define (string-of-char-list? str char-list)
  (string-of-char-list-1? str char-list 0 (string-length str)))

(define (string-of-char-list-1? str char-list i lgt)
  (if (= i lgt)
      #t
      (and (memv (string-ref str i) char-list)
            (string-of-char-list-1? str char-list (+ i 1) lgt))))

;; Are all characters in str different from the characters in char list (a list of characters).
(define (string-of-negative-char-list? str char-list)
  (string-of-negative-char-list-1? str char-list 0 (string-length str)))

(define (string-of-negative-char-list-1? str char-list i lgt)
  (string-of-negative-char-list-1? str char-list (+ i 1) lgt))

--(DOS) -- general.scm 45% L1215 (Scheme) -----
> (string-of-char-list? "abba" (list #\a #\b))
#t
> (string-of-char-list? "aaaa" (list #\a))
#t
> (string-of-char-list? "1 2 3" (list #\1 #\2))
#f
> (string-of-char-list? "1 2 3" (list #\1 #\2 #\3))
#f
> (string-of-char-list? "1 2 3" (list #\1 #\2 #\3 #\space))
#t
>
```


Scheme Mini-IDE

> (string-of-char-list? "abba" (list #\a #\b))

#t

> **OK**

> (string-of-char-list? "1 2 3" (list #\1 #\2))

#f

> **OK**

> (string-of-char-list? #\a (list #\a))

string-length: expects argument of type <string>; given

#\a

> **ERROR**

Scheme Mini-IDE

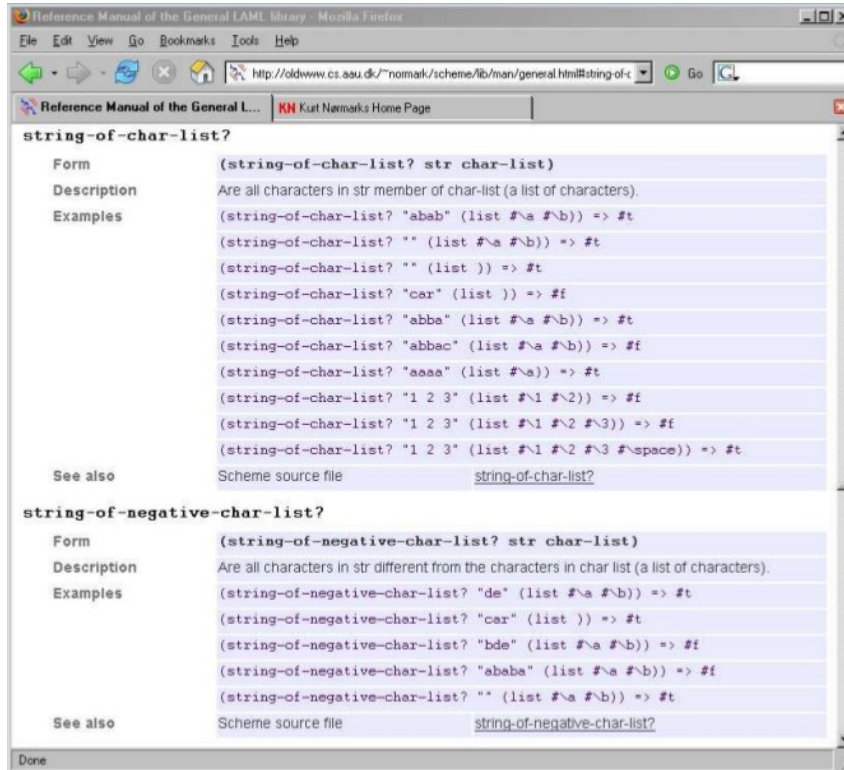
> (string-of-char-list? "abba" (list #\a #\b))
reference to undefined identifier: string-of-char-list?

> **VALUE: #t**

> (string-of-char-list? "abbac" (list #\a #\b))
reference to undefined identifier: string-of-char-list?

> **VALUE: #f**

Scheme Mini-IDE

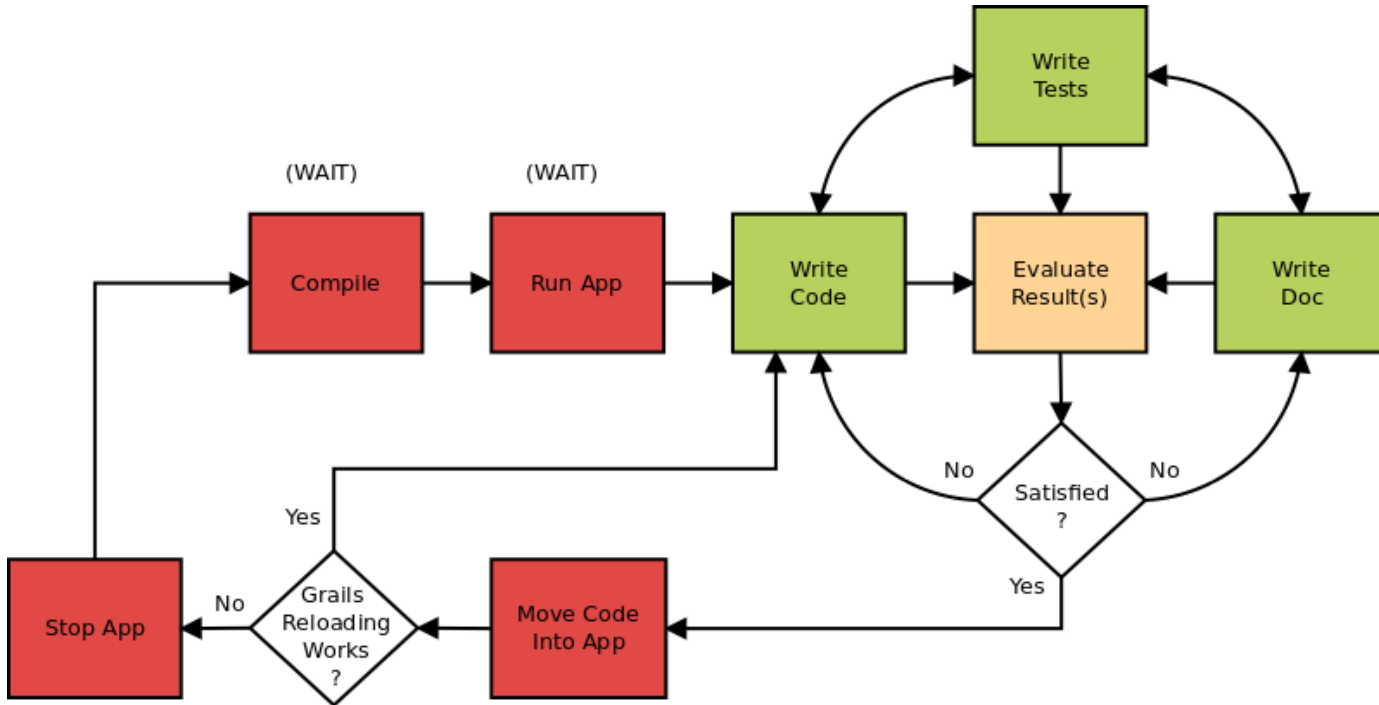


Python Doctest

Example driven, executable doc

```
def square(x):  
    """Squares x.  
  
    >>> square(2)  
    4  
    >>> square(-2)  
    4  
    """  
  
    return x * x
```

The Future?





Thank You For Not Sleeping

(I hope)

Questions?

Links / Credits

- Grails Console Plugin
<http://grails.org/plugin/console>
- Spock Web Console
<https://meetspock.appspot.com/>
- Magnus Rundberget – Groovy Light Table Plugin
<http://codewader.blogspot.no/2014/05/groovy-light-table-client-screencast.html>
- Kurt Nørmarks – Systematic Unit Testing in a Read-eval-print Loop
<http://people.cs.aau.dk/~normark/systematic-unit-test-repl.pdf>
- Python Doctest
<http://docs.python-guide.org/en/latest/writing/tests/>
- Slides
<http://talldave.net>