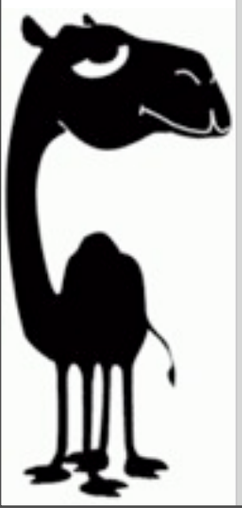
Amsterdam.pm

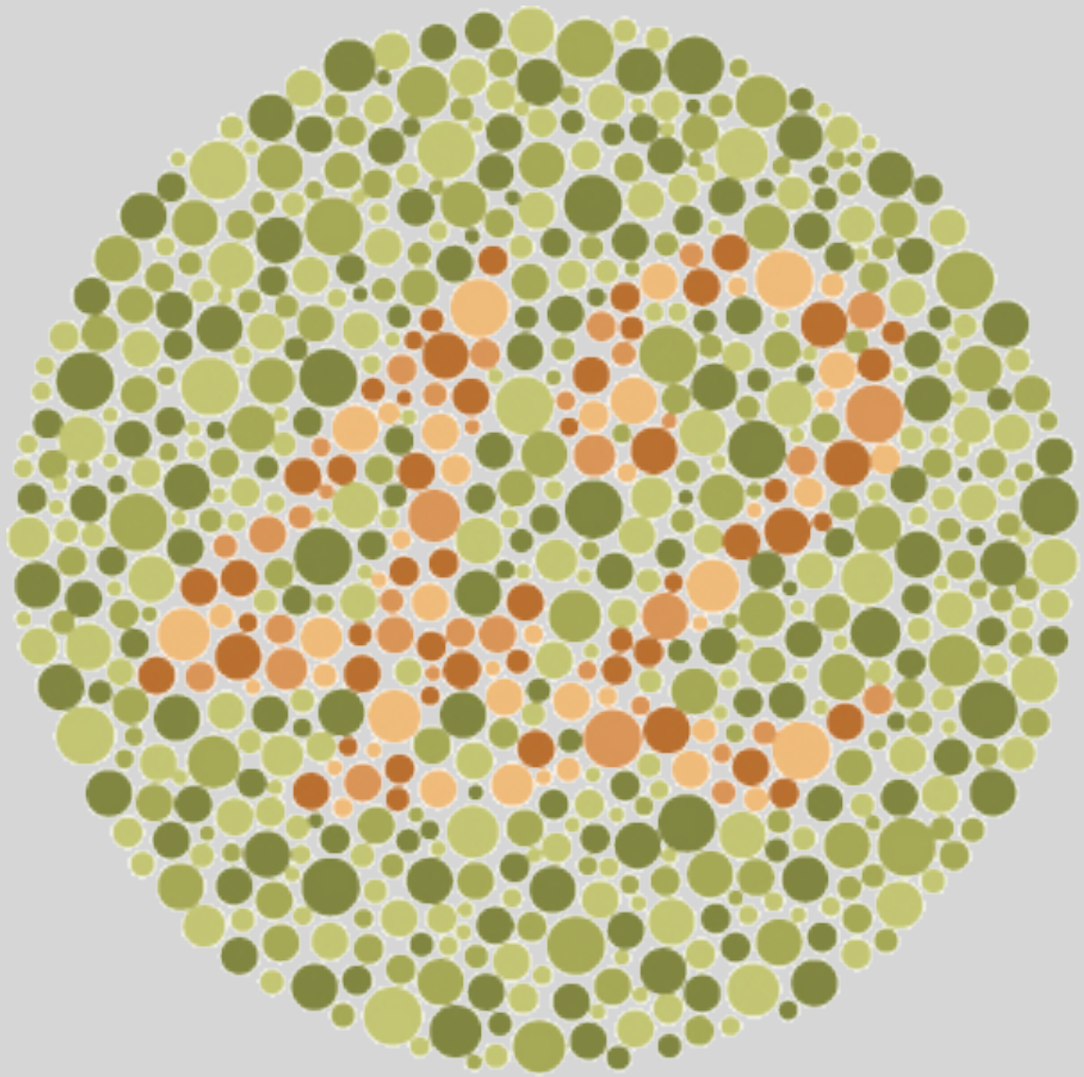
Testing the Tester

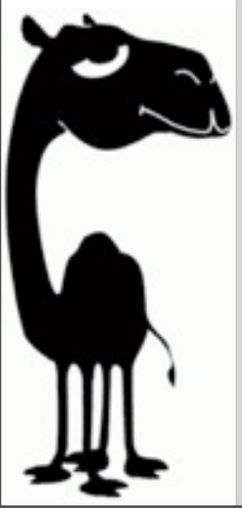
Introduction into Test::Builder::Module and Test::Builder::Tester



```
use YAPC::Europe 2009;
date( '2009-08-04' );
author( abeltje => 'Abe Timmerman' );
```

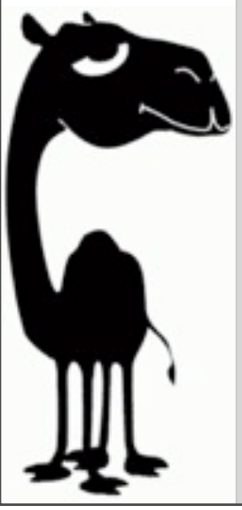
Just testing ;)





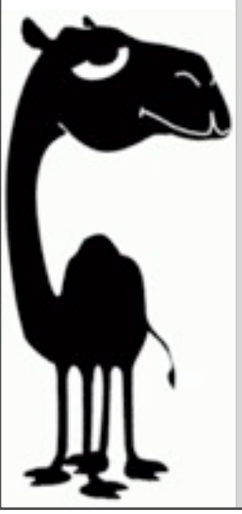
Regression testing

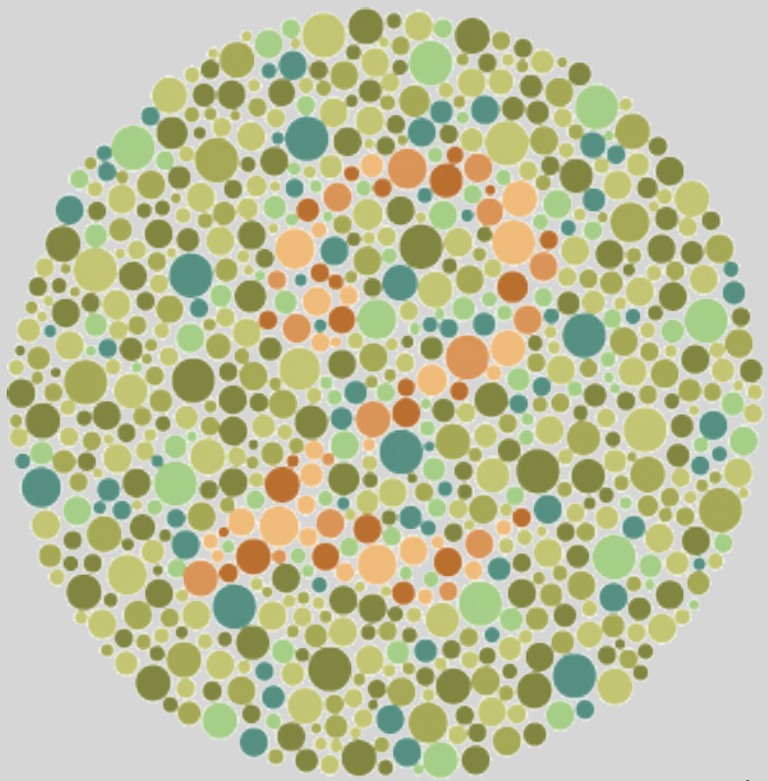
- Found a bug:
  - Testcase (TODO)
  - Commit
- Fixed the bug:
  - Fix testcase (unTODO)
  - Commit



Example (1a)

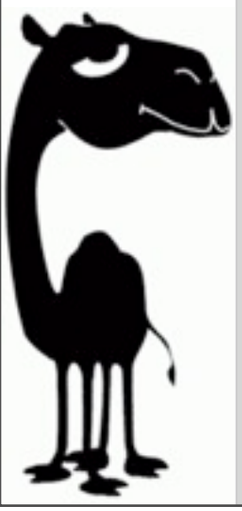
```
#! perl
use warnings;
use strict;

use Test::More tests => 1;
use Test::Exception;

{ # Before the code is fixed
    local $TODO = "bugticket YYY: this dies";
    lives_ok sub {
        code_of_bugreport();
    }, "Just show where the bug is";
}

sub code_of_bugreport {
    die "It really fails.\n";
}
```

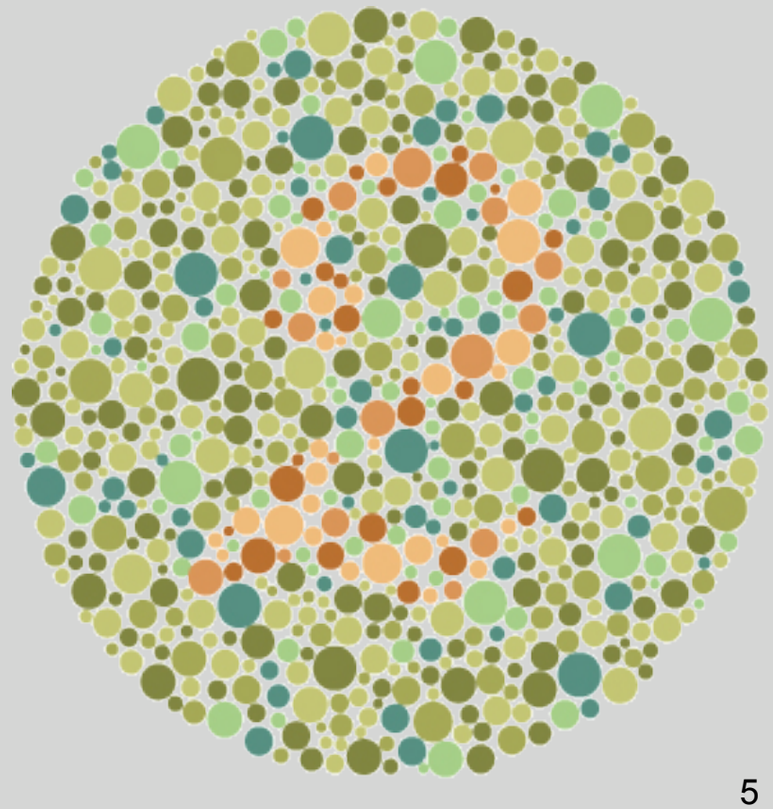




Example (1b)

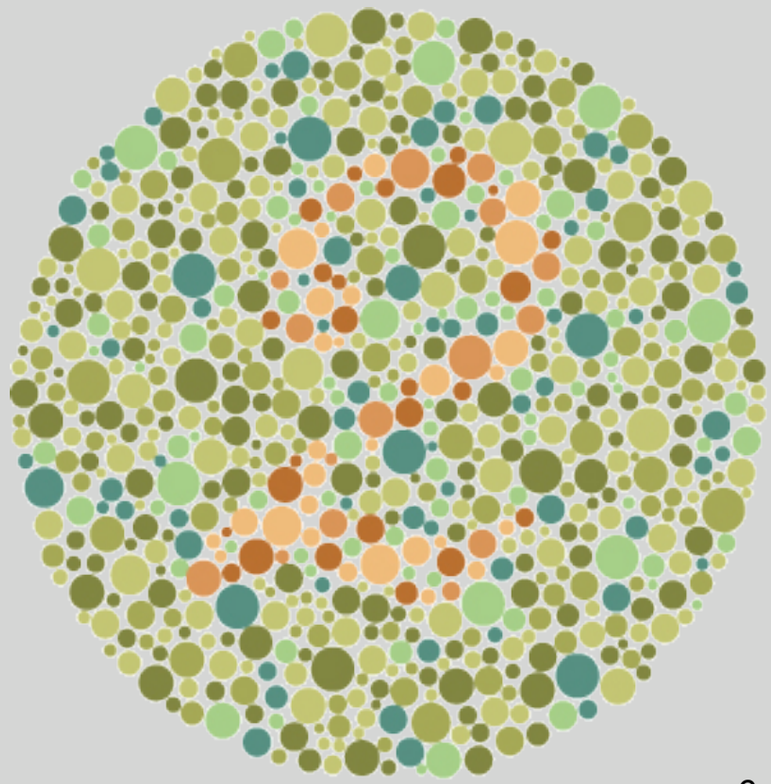
```
#! perl
use warnings;
use strict;

use Test::More tests => 1;
use Test::Exception;

# After the code is fixed
lives_ok sub {
   code_of_bugreport();
}, "Just show where the bug is";

sub code_of_bugreport {
    return "It really passes.\n";
}
```

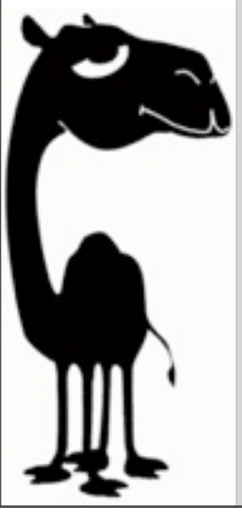




Diff between 1a and 1b

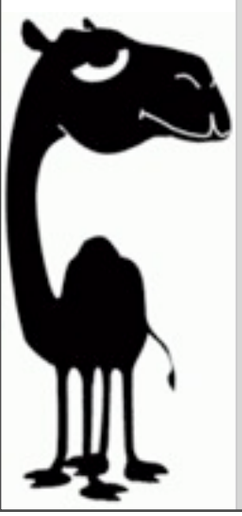
```
--- example1a.pl   2009-06-08 13:34:40.000000000 +0200
+++ example1b.pl   2009-06-08 14:34:42.000000000 +0200
@@ -5,11 +5,8 @@
 use Test::More tests => 1;
 use Test::Exception;

-{
-    local $TODO = "bugticket YYY: this dies";
-    lives_ok sub {
-        code_of_bugreport();
-    }, "Just show where the bug is";
-}
+lives_ok sub {
+    code_of_bugreport();
+}, "Just show where the bug is";
```

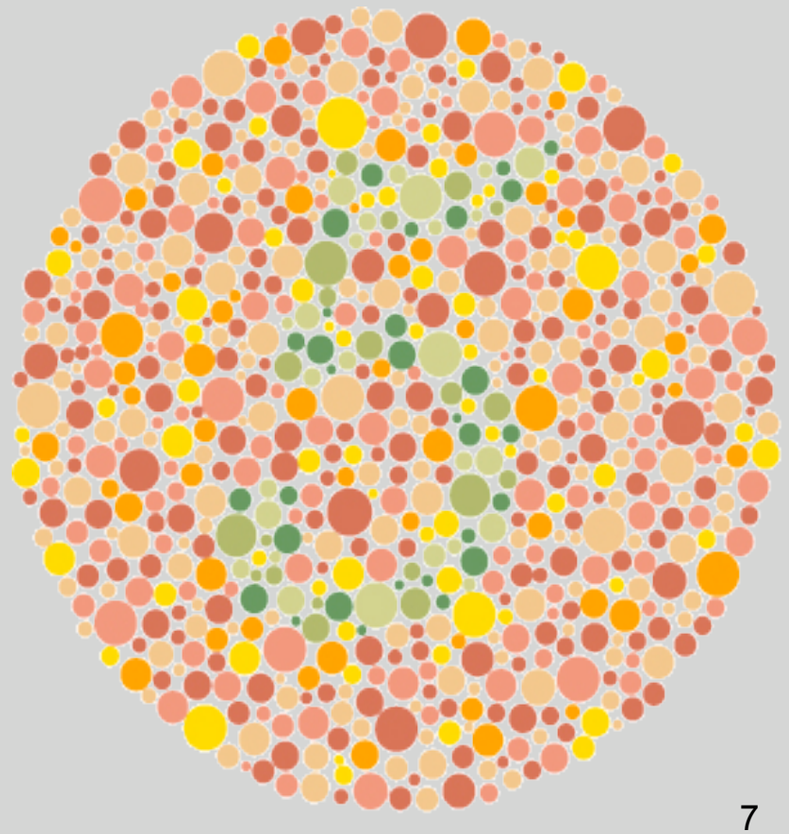




Example (2a)

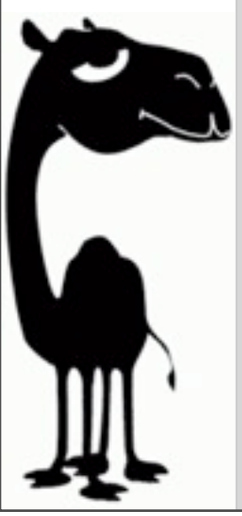
```
#! perl
use warnings;
use strict;

use Test::More tests => 1;
use Test::Regression;

no_regression_ok
    "TODO bugticket YYY: this dies",
    sub {  code_of_bugreport() },
    "Just show where the bug is";

sub code_of_bugreport {
    die "It really fails.\n";
}
```

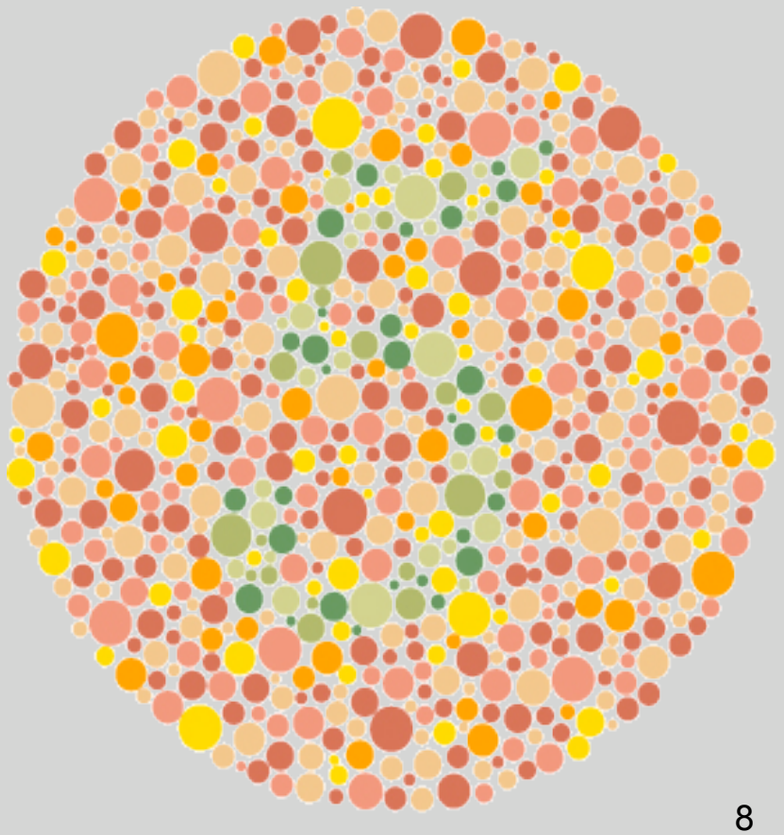




Example (2b)

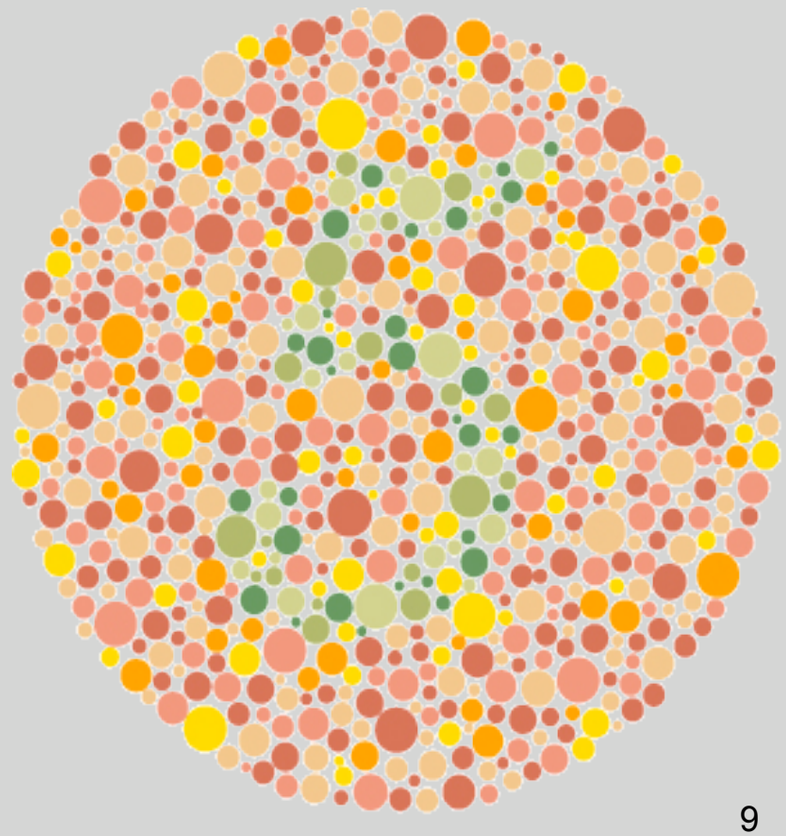
```
#! perl
use warnings;
use strict;

use Test::More tests => 1;
use Test::Regression;

no_regression_ok
    "bugticket YYY: this dies",
    sub {  code_of_bugreport() },
    "Just show where the bug is";

sub code_of_bugreport {
    return "It really passes.\n";
}
```

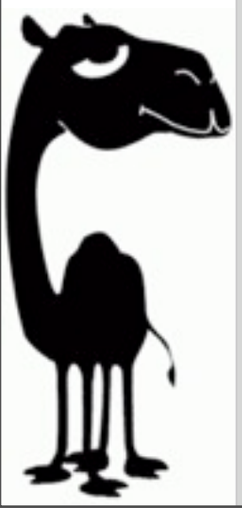




Diff between 2a and 2b

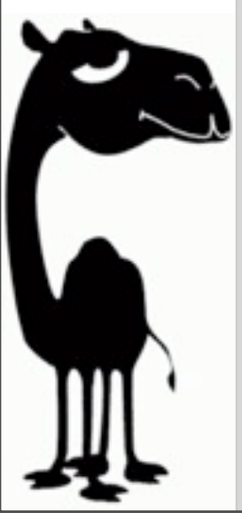
```
--- example2a.pl   2009-06-08 23:48:21.000000000 +0200
+++ example2b.pl   2009-06-09 00:05:15.000000000 +0200
@@ -6,10 +6,10 @@
 use Test::Regression;

 no_regression_ok
-    "TODO bugticket YYY: this dies",
+    "bugticket YYY: this dies",
     sub {  code_of_bugreport() },
     "Just show where the bug is";
```





Your own tester module

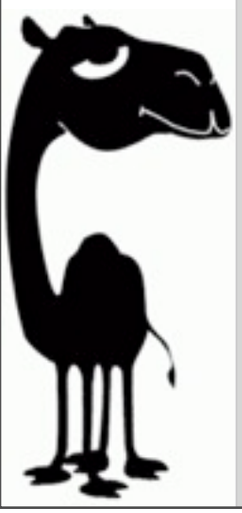
- Subclass Test::Builder::Module
- A Test::Builder instance
  - $tb = Test::Builder::Module->builder
- Use the Test::Builder methods:
  - $tb->ok()
  - $tb->is_eq(), $tb->isnt_eq()
  - $tb->cmp_ok()
  - $tb->like(), $tb->unlike()
  - $tb->diag()



Amsterdam.pm

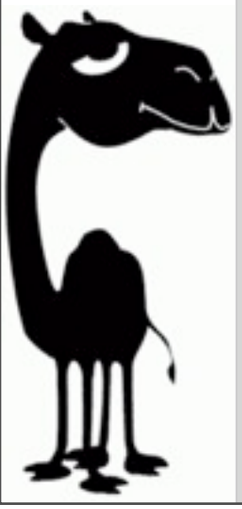
The new testfunction

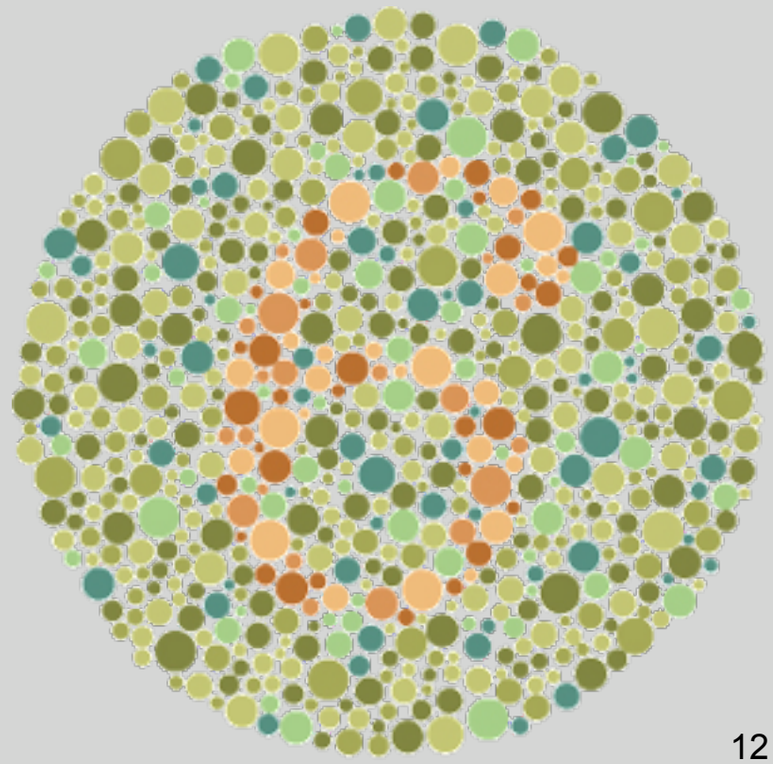
- Create a new tester module:
  - Test::Regression
- Define the new function:
  - no_regression_ok ($&;$)



Test::Regression (1)

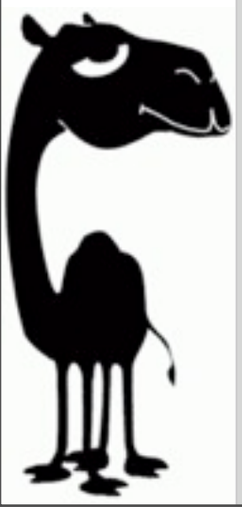
```
package Test::Regression;
use warnings;
use strict;

use base 'Test::Builder::Module';
use Exporter 'import';
our @EXPORT = qw/ no_regression_ok /;

my $tb = __PACKAGE__->builder;
```

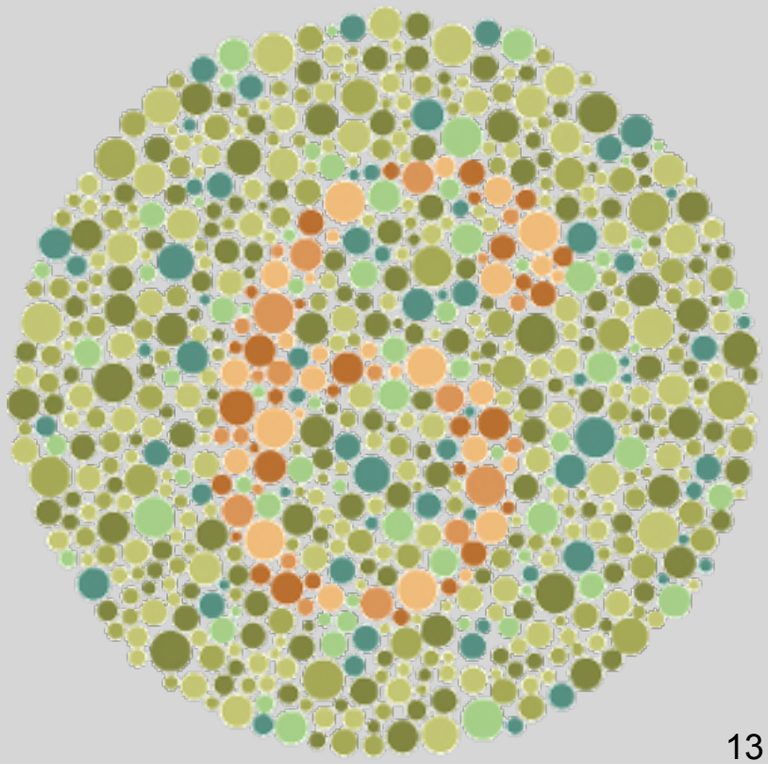




Test::Regression (2)

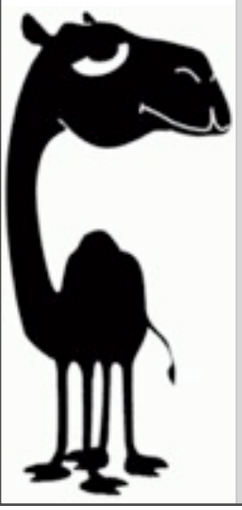
```
sub no_regression_ok ($&;$) {
    my ($todo, $code, $msg ) = @_;
    defined $todo or $todo = "";
    defined $msg  or $msg  = "";

    local $::TODO; # Test::Builder looks for $main::TODO
    if ( $todo =~ s/^TODO(?:\s|$)// ) {
        $::TODO = $todo || 'XXX';
    }
    else {
        $msg .= " $todo";
    }

    my $result = eval { $code->() };
    my $error = $@;
    if ( $error ) {
        $tb->diag( "[Regression DIE] $error" );
        return $tb->ok( 0, $msg );
    }
    if ( ! $result ) {
        $tb->diag( "[Regression FAIL]" );
        return $tb->ok( 0, $msg );
    }

    return $tb->ok( 1, $msg );
}
```





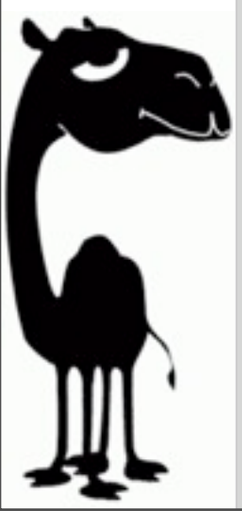
Test::Builder::Tester

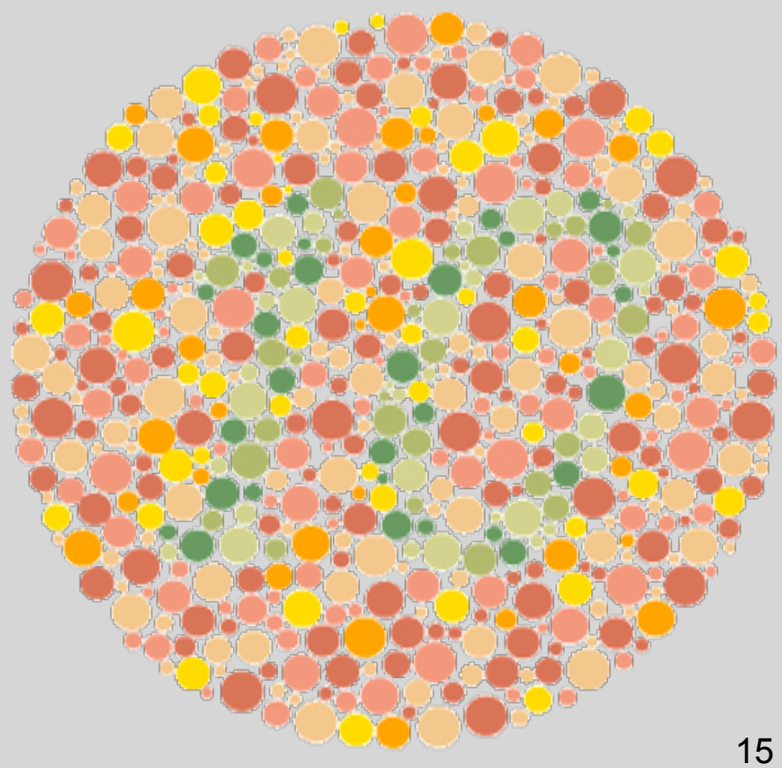
- For each test:
  - Predict the output:
    - STDOUT: test_out()
    - STDERR: test_err()
    - Fail shortcut: test_fail()
  - Call the testfunction
  - Check the result: test_test()



Testing Test::Regression (1)

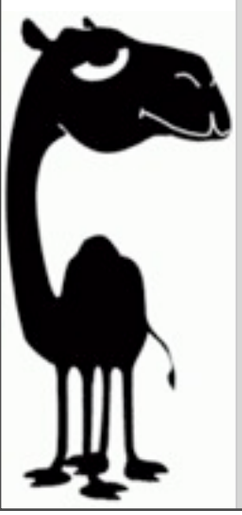
```
#! perl
use warnings;
use strict;

use Test::Builder::Tester tests => 4;
use Test::Regression;

test_out "ok 1 - extra info";
no_regression_ok "extra info", sub { 1 };
test_test "pass basic test with extra info";

test_out "ok 1 - basic test extra info";
no_regression_ok "extra info", sub { 1 }, "basic test";
test_test "pass basic test with name and extra info";

test_out "ok 1 - basic test # TODO todotest";
no_regression_ok "TODO todotest", sub { 1 }, "basic test";
test_test "pass basic test with TODO";

test_out "ok 1 - basic test # TODO XXX";
no_regression_ok "TODO", sub { 1 }, "basic test";
test_test "pass basic test with TODO XXX";
```

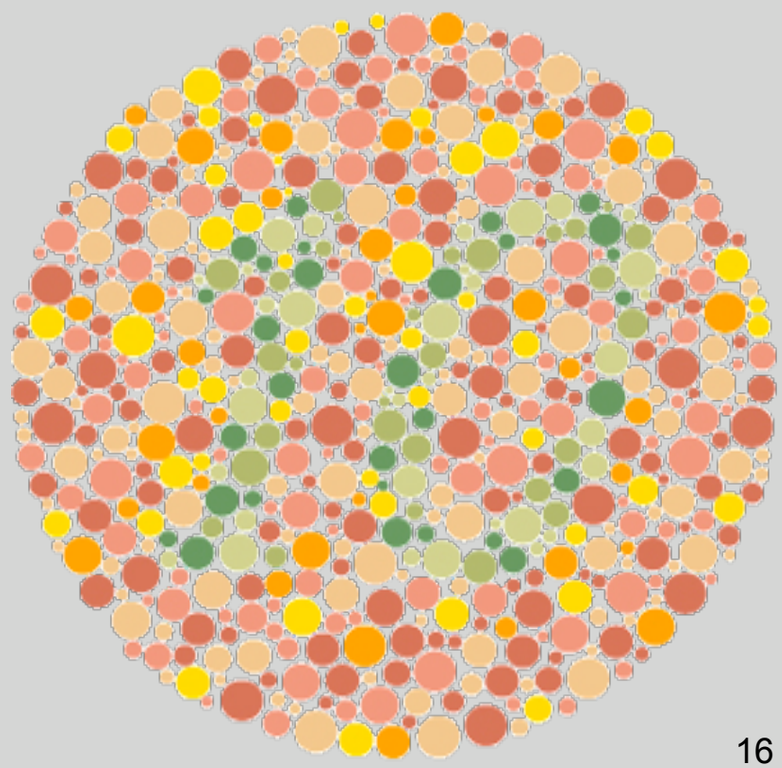




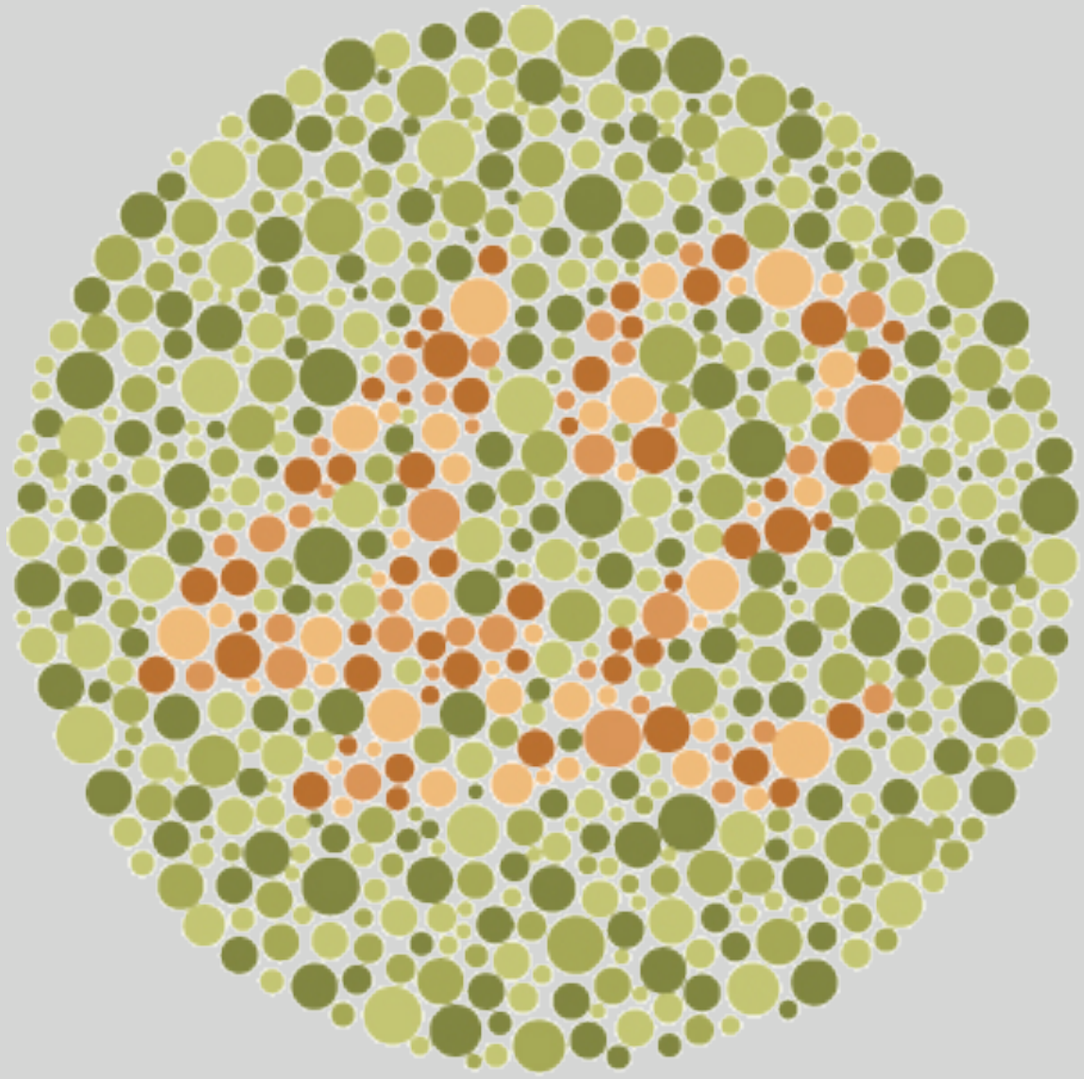
Testing Test::Regression (2)

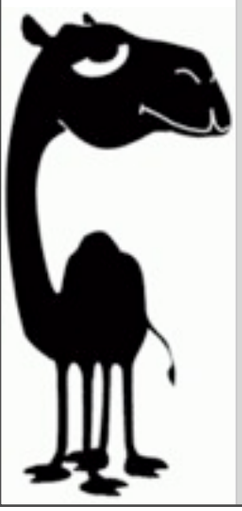
```
#! perl
use warnings;
use strict;

use Test::Builder::Tester tests => 2;
use Test::Regression;

test_out "not ok 1 - basic test # TODO todotest";
test_err "# [Regression FAIL]";
test_err qr/#\s+Failed \(TODO\) test.*?\n?.*?at $0 line \d+.\n/;
no_regression_ok
    "TODO todotest",
    sub { 0 },
    "basic test";
test_test "failed basic test with TODO";

test_out "not ok 1 - basic test # TODO todotest";
test_err "# [Regression DIE] arg";
test_err qr/#\s+Failed \(TODO\) test.*?\n?.*?at $0 line \d+.\n/;
no_regression_ok
    "TODO todotest",
    sub { die "arg\n" },
    "basic test";
test_test "died basic test with TODO";
```





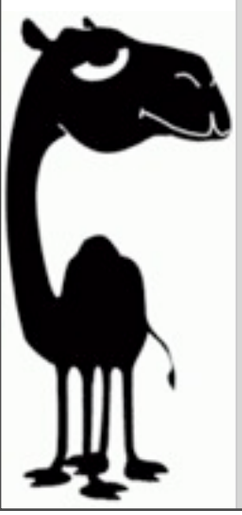
Bonus





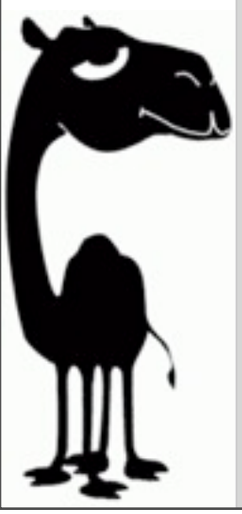
Drop-in replacment for Test::More

- Use your favorite Test::* modules
  - Test::More
  - Test::Exception
  - Test::Warn
  - Test::NoWarnings
- Add your own testerfunctions
- Clean-up your test-suite!



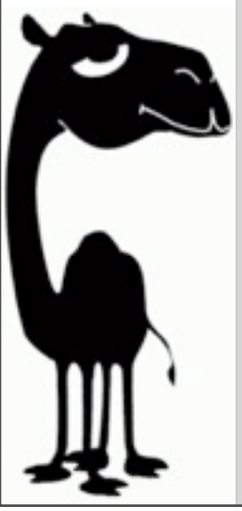
Test::Kit

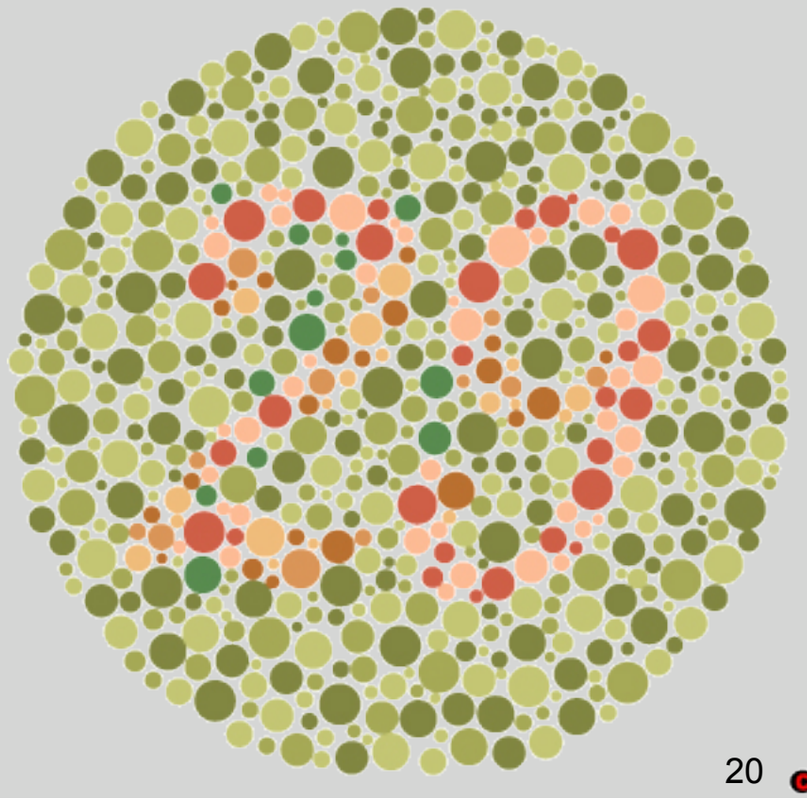
- A safe way to replace Test::More
- Extra feature
  - on_fail()
- Easy to use
- Caveat: still α



Test::NoRegression

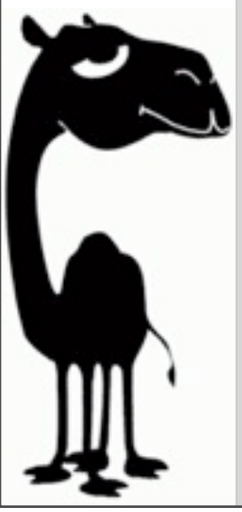
```
package Test::NoRegression;
use warnings;
use strict;

use Test::Kit qw{
    Test::More
    Test::Warn
    Test::Exception
    Test::NoWarnings
    Test::Regression
};

1;
```

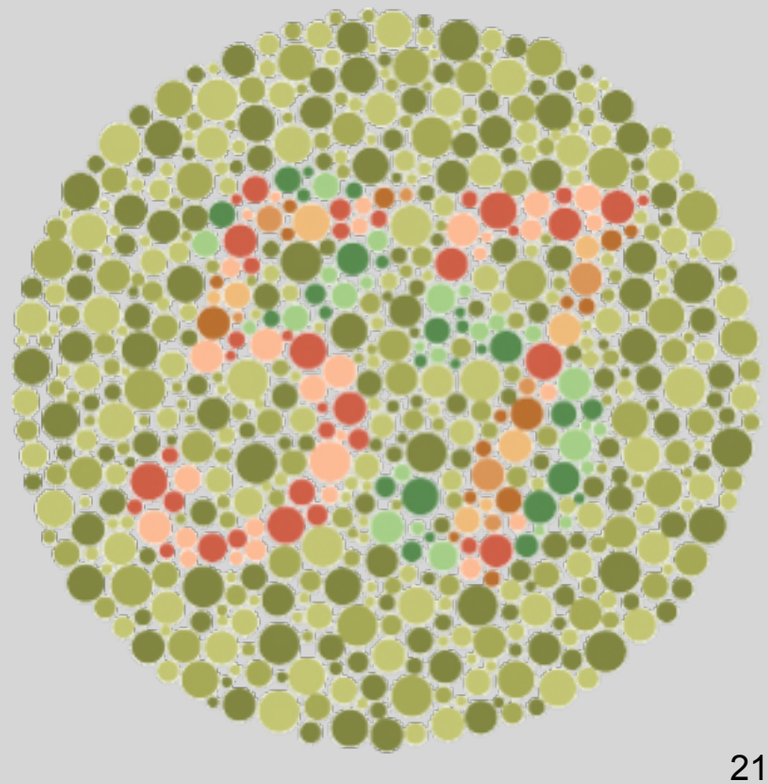




Example usage

```
#! perl
use warnings;
use strict;

use Test::NoRegression tests => 2;

no_regression_ok
    "TODO bugticket YYY: this dies",
    sub {  code_of_bugreport() },
    "Just show where the bug is";

sub code_of_bugreport {
    die "It really fails.\n";
}
```





Questions?

Thank you!


Thank you!
謝謝
Dank Je Wel!
GRACIAS!
Merci!
Todah!
Danke!
Gracias!
Thank you!
どうもありがとうございました
Dank Je Wel!
ThANK YoU!
Merci!
Todah!
GRACIAS!
感謝
MERCI!