QuickCheck Evolution

John Hughes

CHALMERS QuviQ

Why is testing hard?

$O(n^3)$ test cases

3—4 tests per pairs of features

triples of features

feature

Don’t write tests!

Generate them

QuickCheck

1999—invented by Koen Claessen and myself, for Haskell

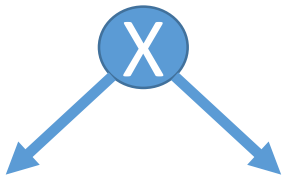
2006—Quviq founded marketing Erlang version

Many extensions

Finding deep bugs for Ericsson, Volvo Cars, Basho, etc…

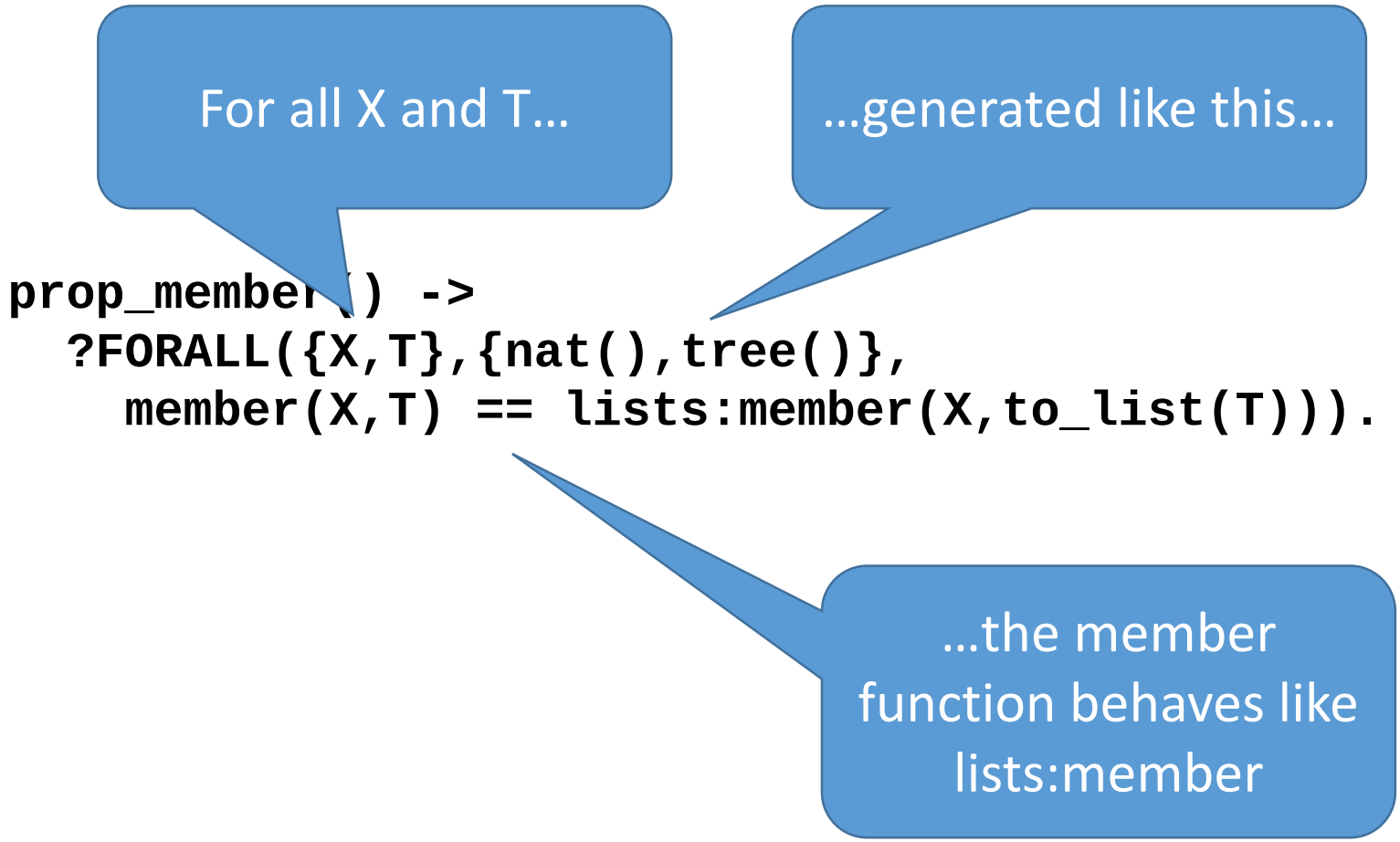
Example—binary trees

`leaf` 

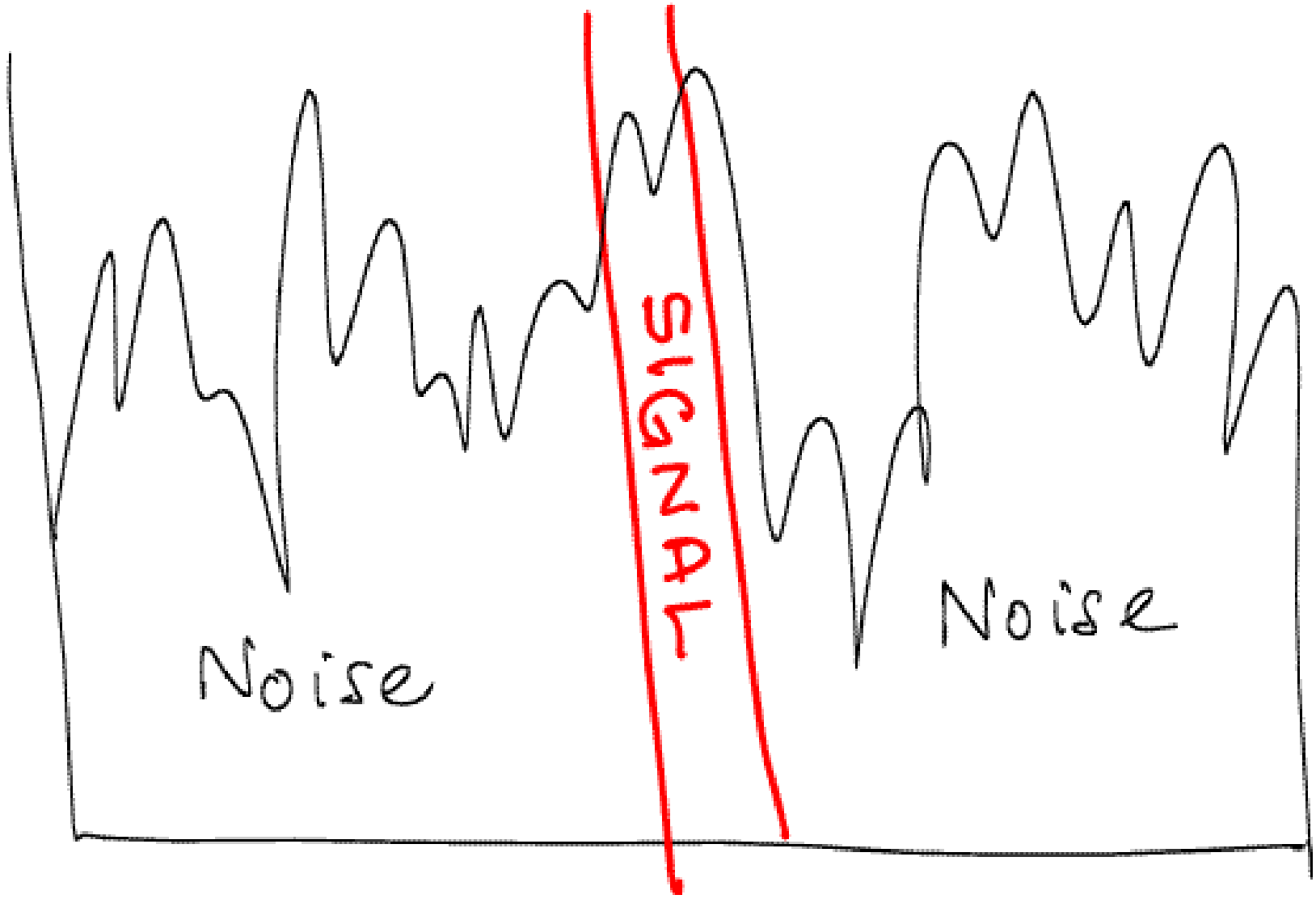
`{node,L,X,R}` 


```
to_list(leaf) -> [];
to_list({node,L,X,R}) ->
  to_list(L) ++ [X] ++ to_list(R).

member(_,leaf) ->
  false;
member(X,{node,L,Y,R}) ->
  if X<Y  -> member(X,R);
     X==Y -> true;
     X>Y  -> member(X,L)
  end.
```

A property of member



Let’s run some tests…


SIGNAL
Noise
Noise

But… what was that example again?

- We may want to *preserve* examples that failed before, as a regression suite

- In reality, a failing case may take a long time to find… we don’t want to throw it away!

Enter… QuickCheck CI

DEMO

VIRTUAL


QuickCheck | Contin
quickcheck-ci.com
QuviQ
Home
Find project
Register project
Help
quickcheck-ci.com
Welcome to QuickCheck CI!
QuickCheck CI is a continuous integration server that runs QuviQ QuickCheck on a project. QuickCheck is a tool for automatically generating test cases from specified properties of the software under test.
Open Source Developers can use QuickCheck CI to get free access to QuviQ QuickCheck for their quality assurance. There are many open source projects created by people that have a commercial copy of QuickCheck; they can now easily share their QuickCheck properties. QuickCheck CI runs the full version of QuickCheck, including the connection to C. Thus, even open source projects written in C can be tested with QuickCheck CI.
Register your
Find registered project

Project instructions

Required files

Before registering your project you need to add the following two files in the top directory of your repository:

⇒ `EQC_CI_LICENCE.txt` - The QuviQ CI licence file which can be downloaded here.

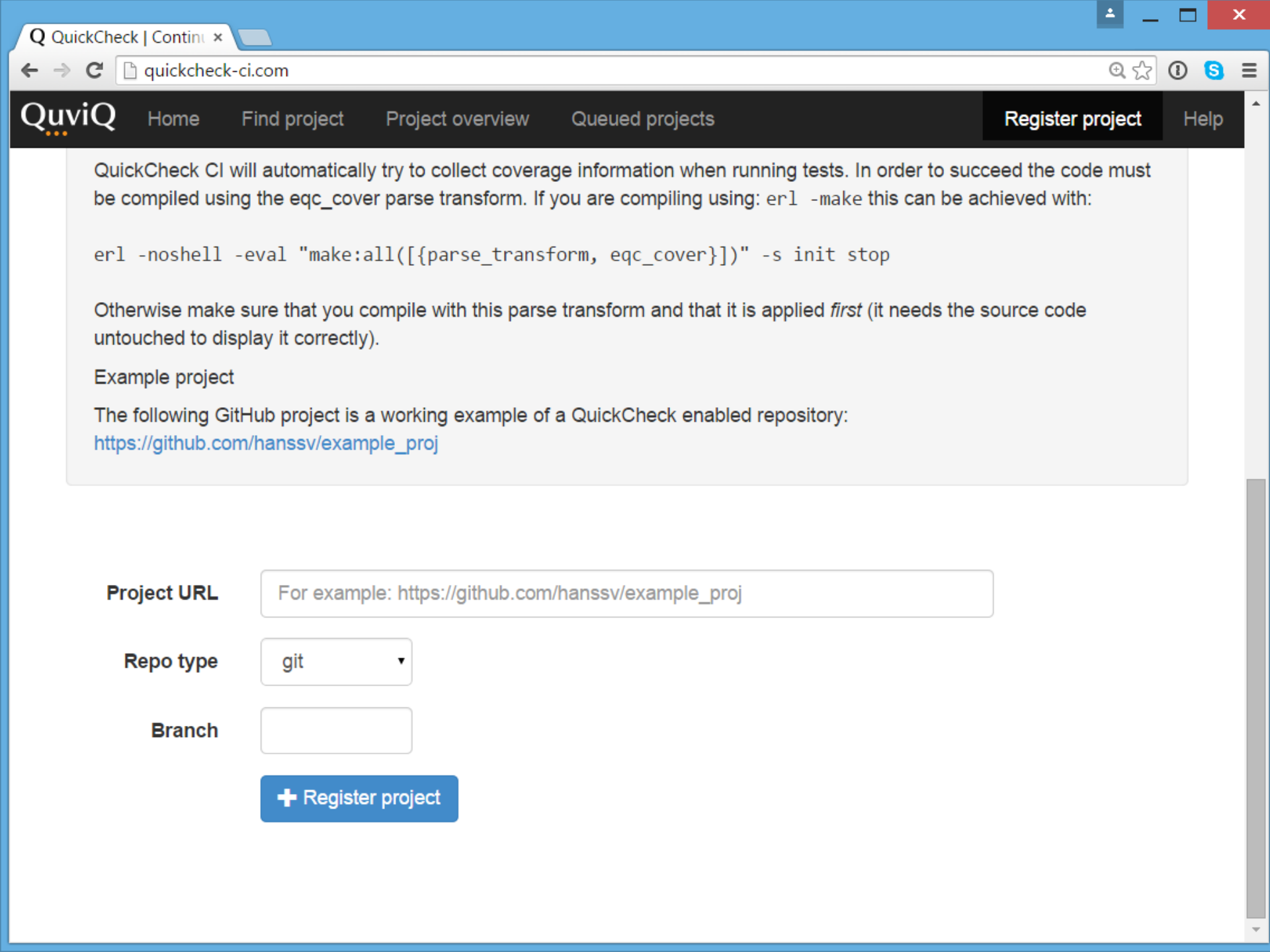
⇒ `.eqc_ci` - A configuration file with the following contents.

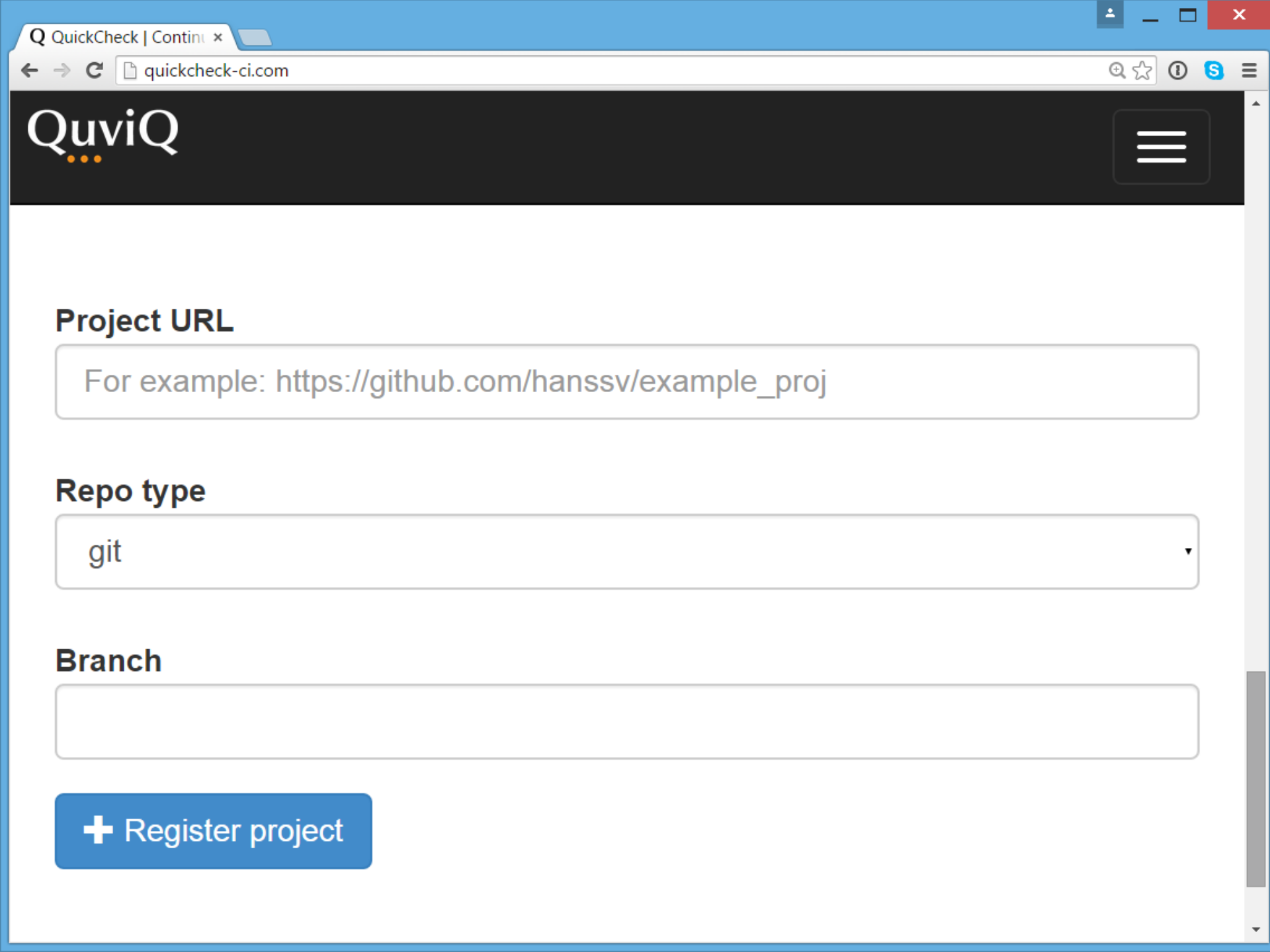
`{build, "CMD"}.`	Where `CMD` is a command that builds your project, for example: `cd ebin; erl -make` (*Mandatory*)
`{test_path, "DIR"}.`	Where `DIR` is the directory where the compiled **beam**-files will be found, for example: `ebin` (*Optional* - the option can be repeated if tests reside in multiple directories. *Default:* `ebin`)
`{deps, "DIR"}.`	Where `DIR` is the directory containing beam files necessary when running, for example: `deps/folsom/ebin` (*Optional* - the option can be repeated for multiple dependencies. **Note:** rebar.config is currently not taken into account)
`{test_root, "DIR"}.`	Where `DIR` is the directory from where the tests should be run, for example: `ebin_test` (*Optional* - the option can be repeated for multiple dependencies. **Note:** rebar.config is currently not taken into account)

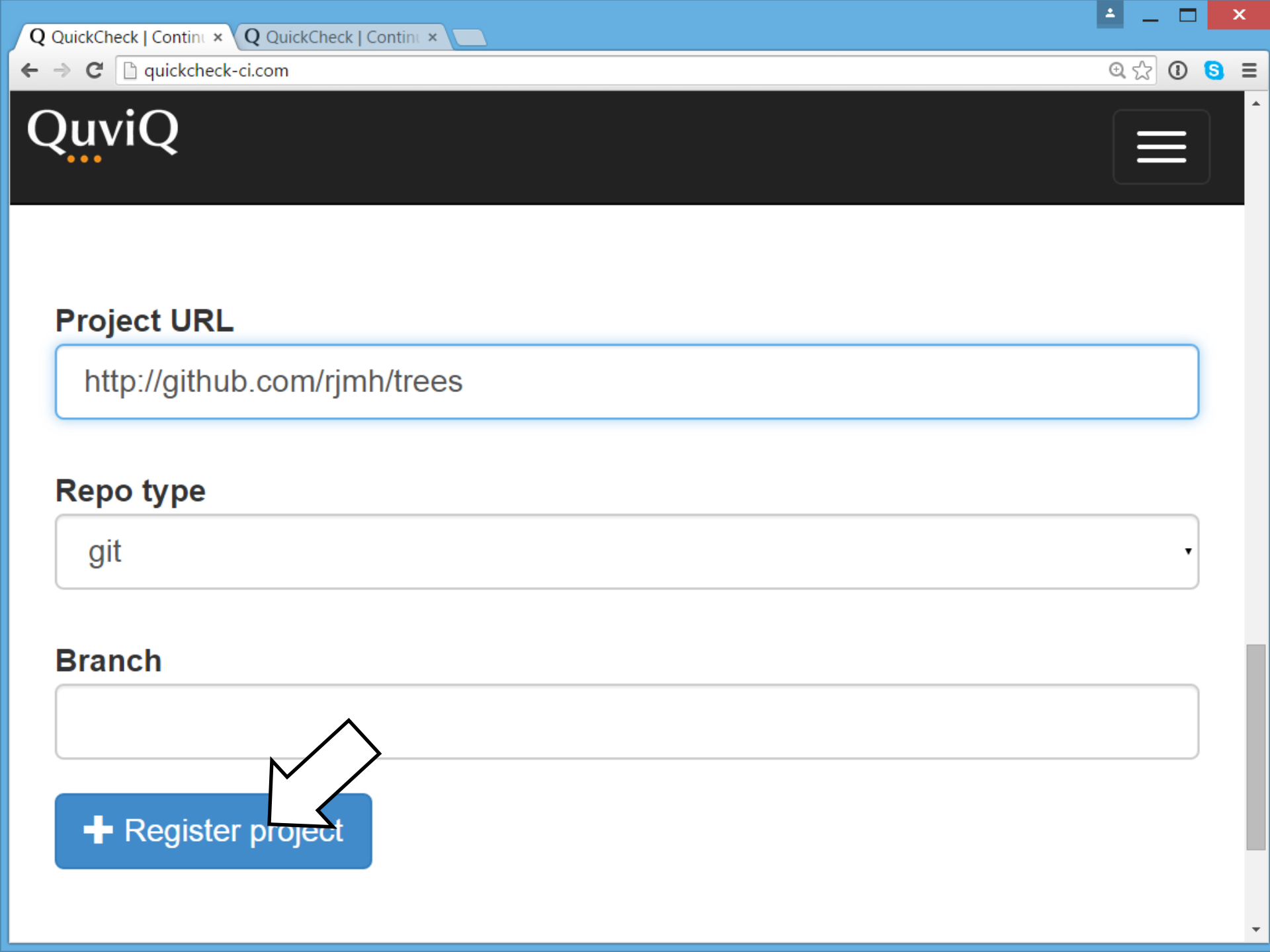
Code coverage

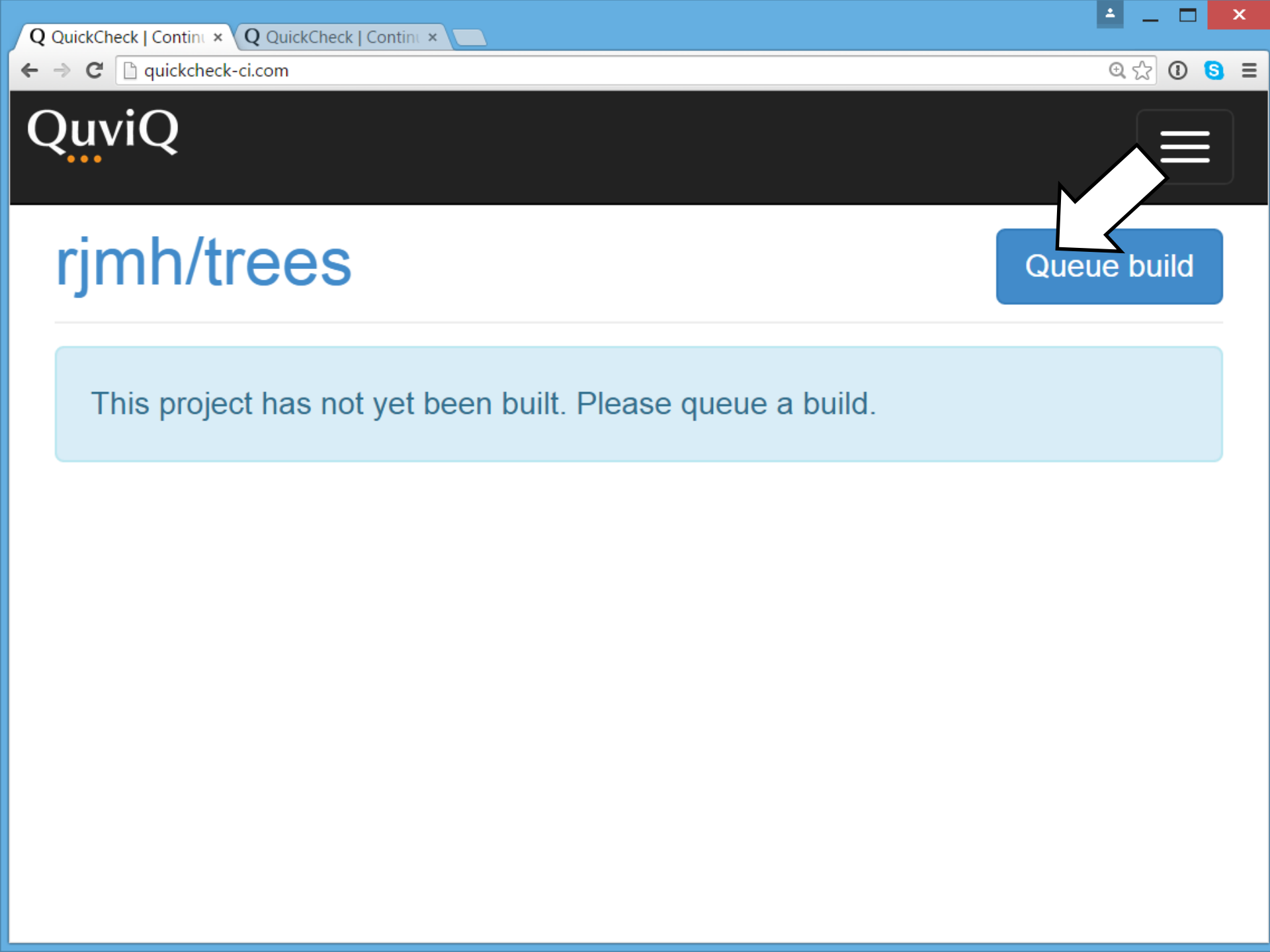
QuickCheck CI will automatically try to collect coverage information when running tests. In order to succeed the code must be compiled using the eqc_cover parse transform. If you are compiling using: `erl -make` this can be achieved with:

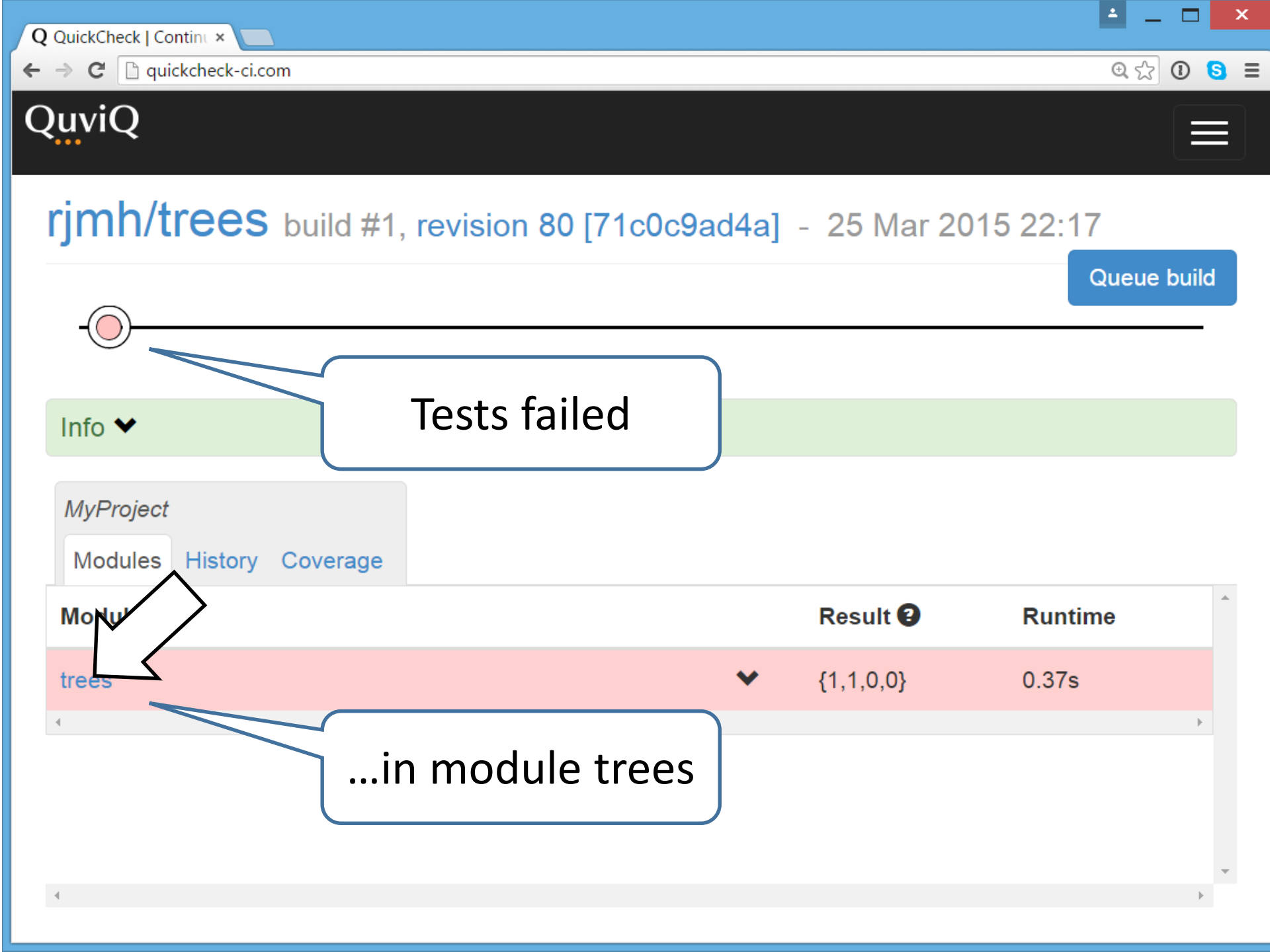
```
erl -noshell -eval "make:all([{parse_transform, eqc_cover}])" -s init stop
```


QuickCheck | Contin
quickcheck-ci.com
QuviQ
Home
Find project
Project overview
Queued projects
Register project
Help
QuickCheck CI will automatically try to collect coverage information when running tests. In order to succeed the code must be compiled using the eqc_cover parse transform. If you are compiling using: erl -make this can be achieved with:
erl -noshell -eval "make:all([{parse_transform, eqc_cover}])" -s init stop
Otherwise make sure that you compile with this parse transform and that it is applied first (it needs the source code untouched to display it correctly).
Example project
The following GitHub project is a working example of a QuickCheck enabled repository:
https://github.com/hanssv/example_proj
Project URL
For example: https://github.com/hanssv/example_proj
Repo type
git
Branch
Register project


QuickCheck | Contin
quickcheck-ci.com
QuviQ
Project URL
For example: https://github.com/hanssv/example_proj
Repo type
git
Branch
Register project

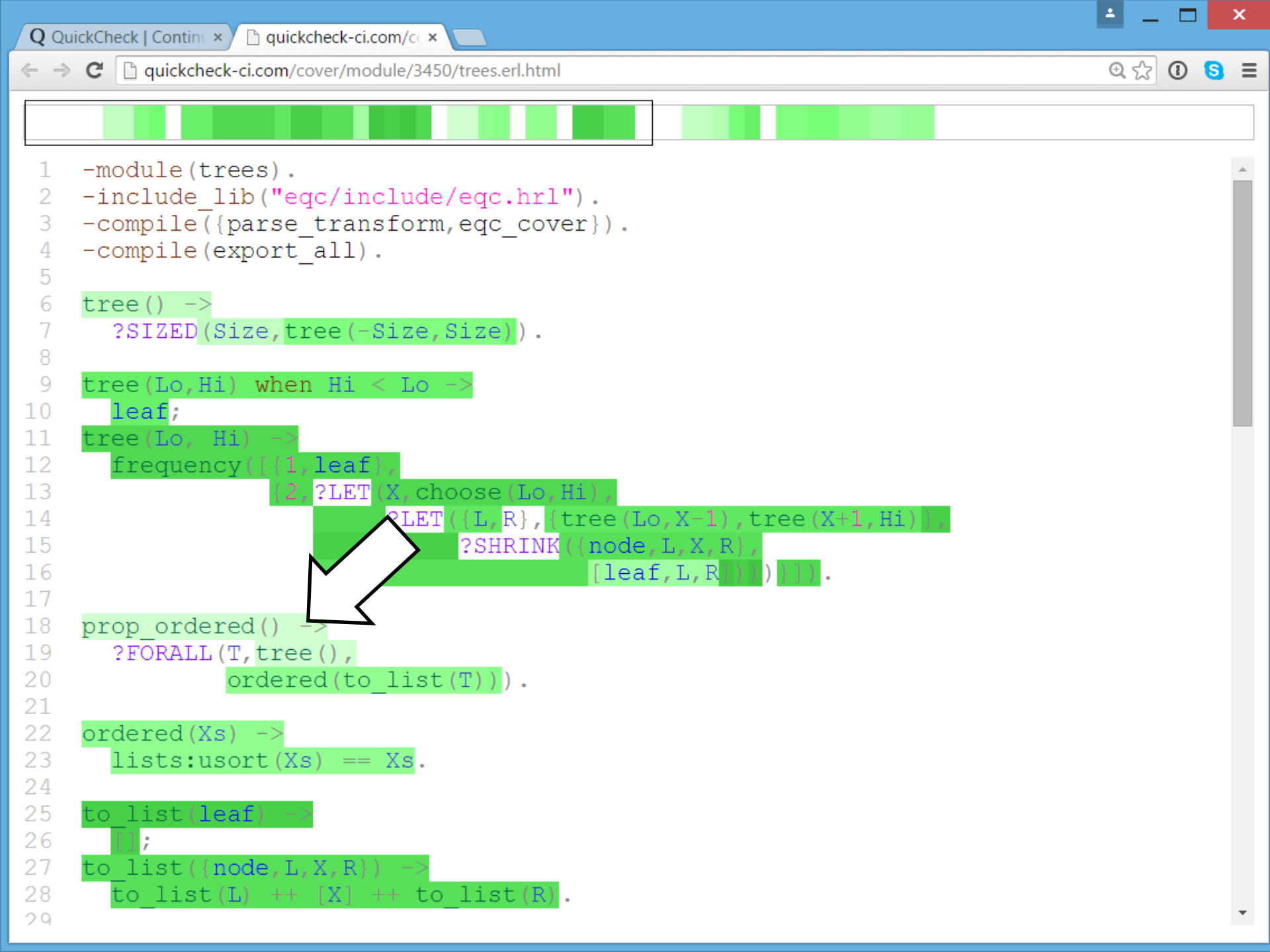

QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
Project URL
http://github.com/rjmh/trees
Repo type
git
Branch
Register project

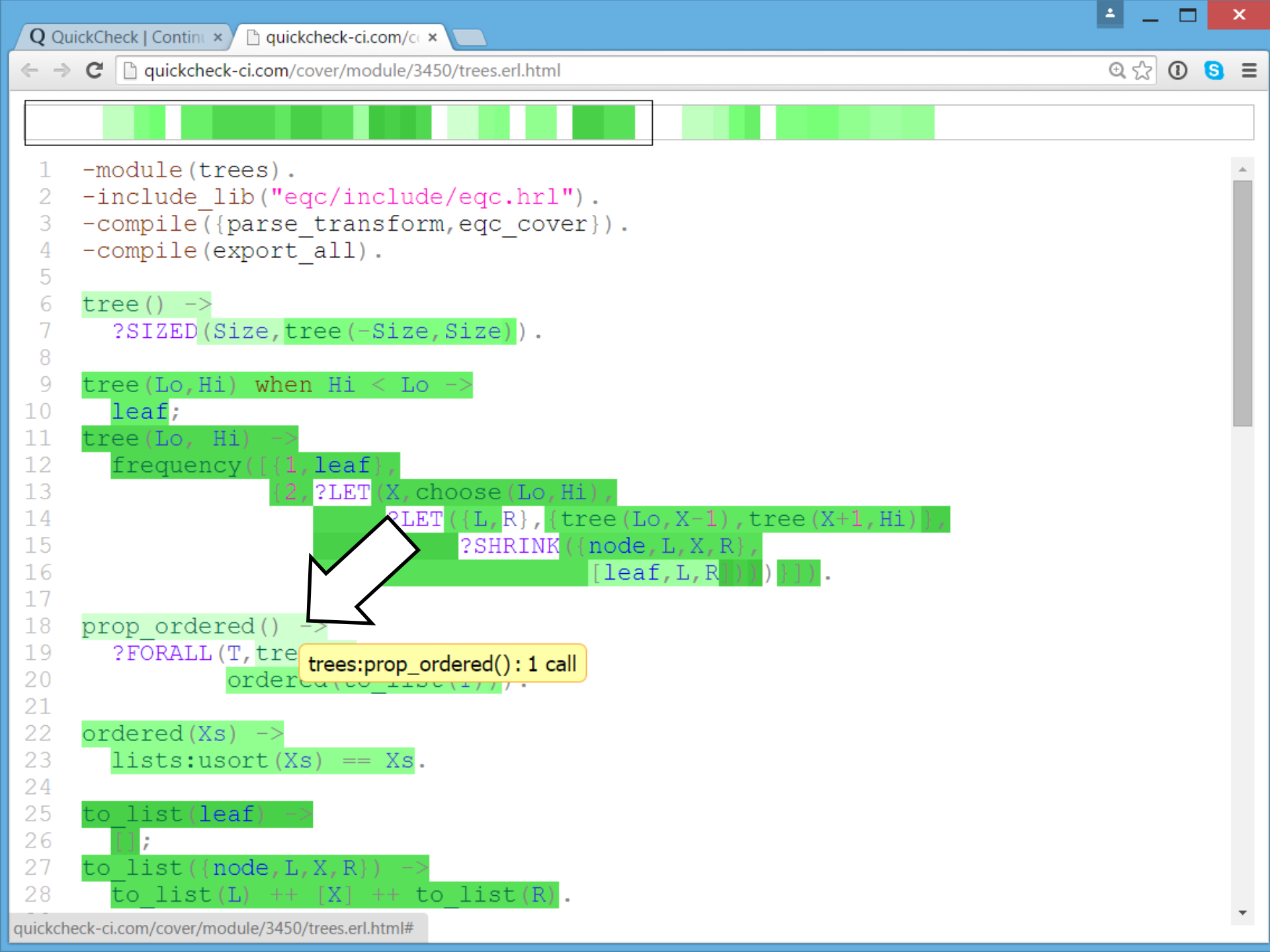

QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/trees
Queue build
This project has not yet been built. Please queue a build.

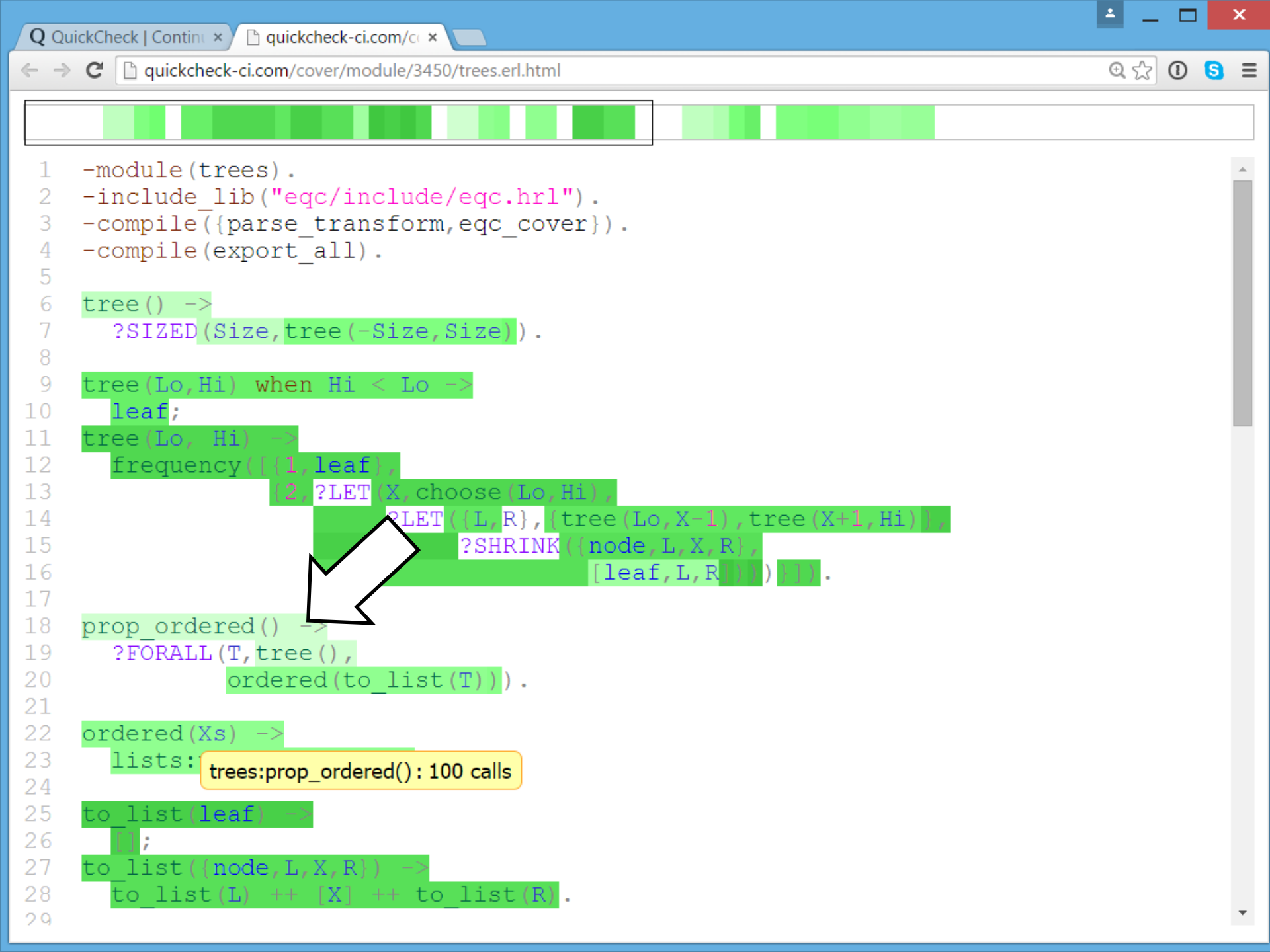

QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/trees build #1, revision 80 [71c0c9ad4a] - 25 Mar 2015 22:17
Queue build
Tests failed
Info
MyProject
Modules
History
Coverage
Result
Runtime
trees
{1,1,0,0}
0.37s
...in module trees

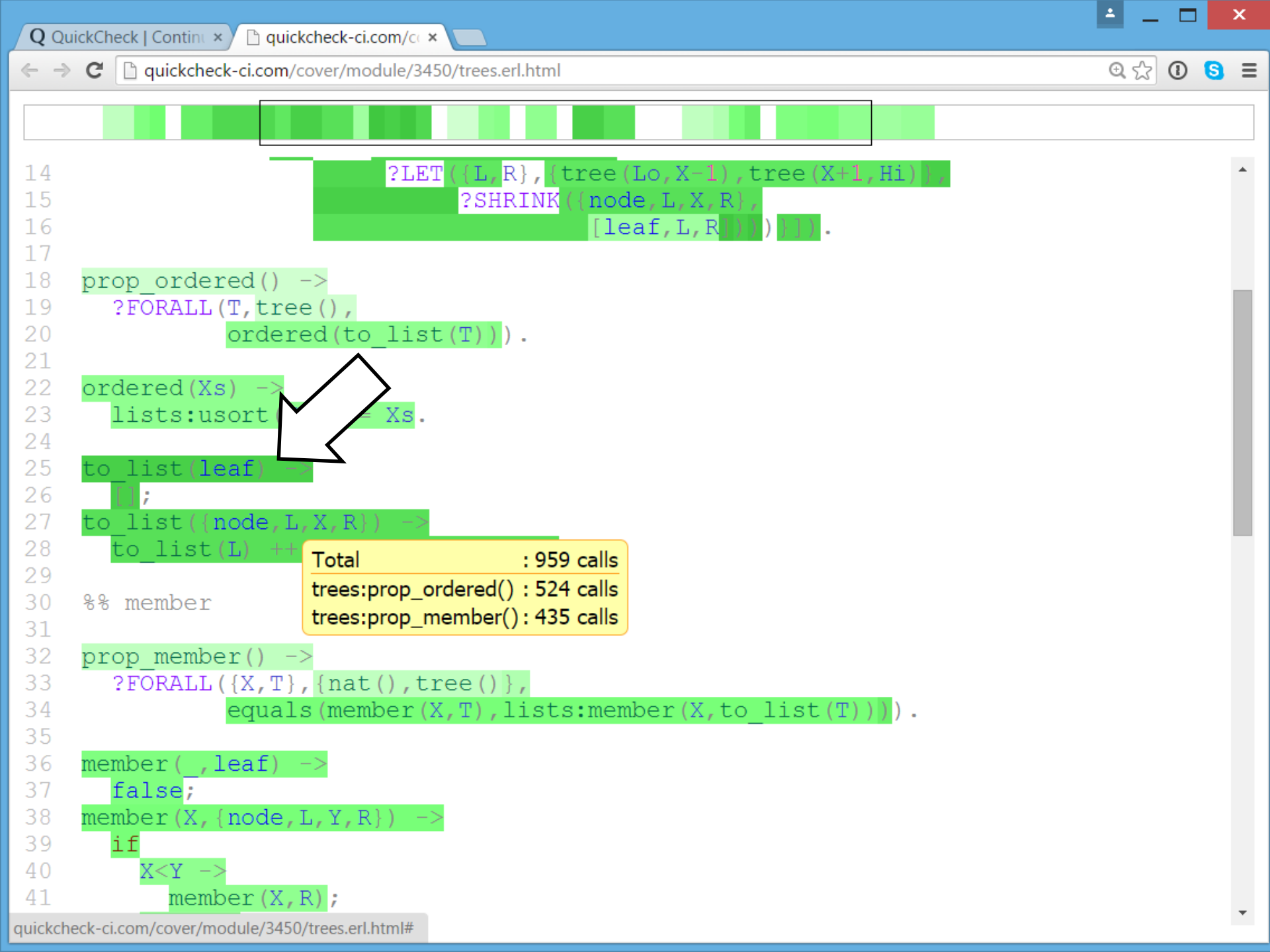

QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/trees build #1, revision 80 [71c0c9ad4a] - 25 Mar 2015 22:17
Queue build
Info
MyProject
Modules History Coverage
trees
Properties History Coverage
Property
Output
Numtests
Runtime
prop_member
Shrinking (0 times) [+]
9
0.27s
prop_ordered
OK, passed 100 tests
100
0.12s

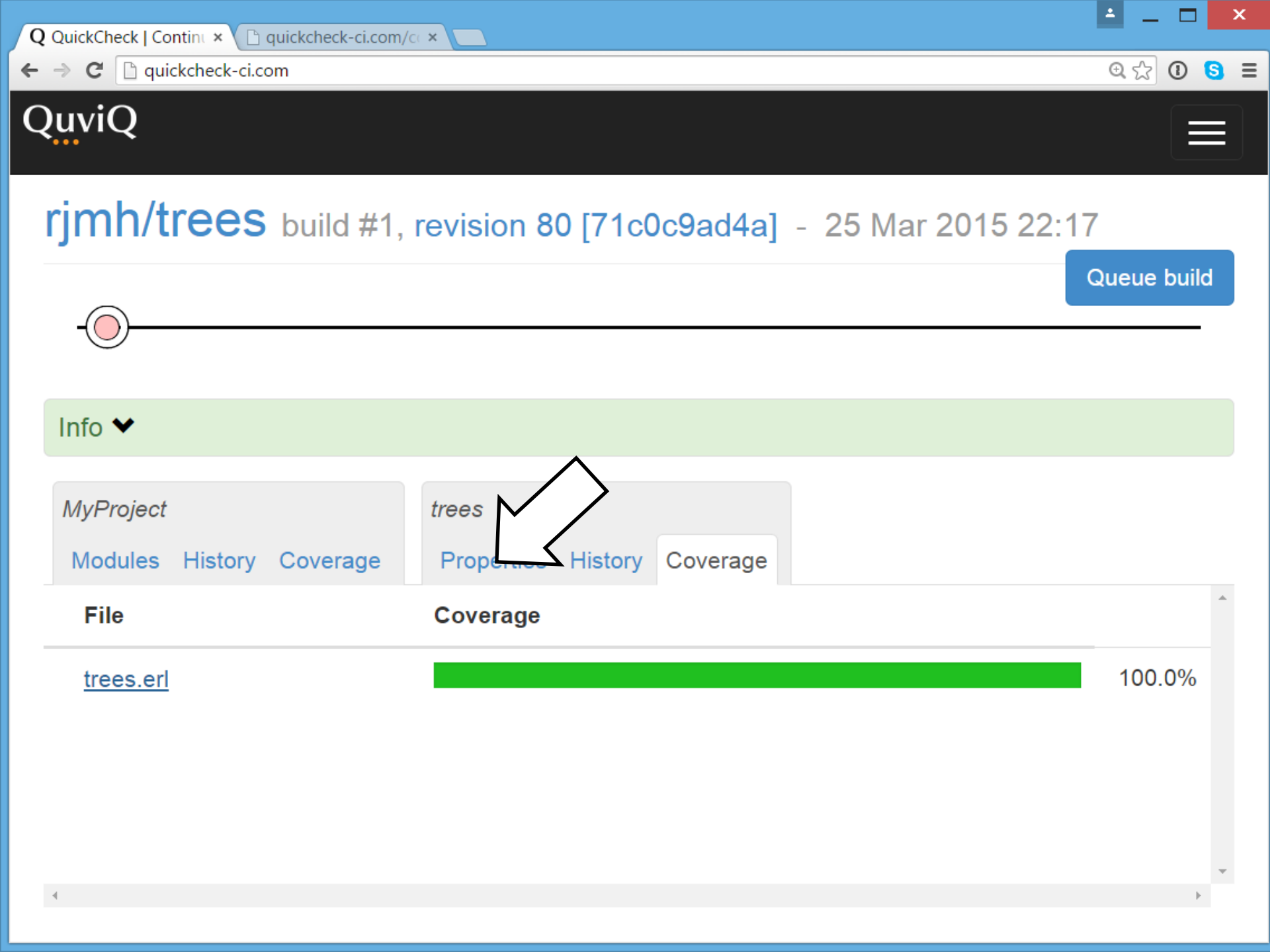

QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/trees build #1, revision 80 [71c0c9ad4a] - 25 Mar 2015 22:17
Queue build
Info
MyProject
Modules History Coverage
trees
Properties History Coverage
File
Coverage
trees.erl
100.0%
quickcheck-ci.com/#module_coverage


QuickCheck | Contin
quickcheck-ci.com/c
quickcheck-ci.com/cover/module/3450/trees.erl.html
1 -module(trees).
2 -include_lib("eqc/include/eqc.hrl").
3 -compile({parse_transform,eqc_cover}).
4 -compile(export_all).
5
6 tree() ->
7 ?SIZED(Size,tree(-Size,Size)).
8
9 tree(Lo,Hi) when Hi < Lo ->
10 leaf;
11 tree(Lo, Hi) ->
12 frequency([{1,leaf},
13 {2,?LET(X,choose(Lo,Hi),
14 ?LET({L,R},{tree(Lo,X-1),tree(X+1,Hi)},
15 ?SHRINK({node,L,X,R},
16 [leaf,L,R]))}]).
17
18 prop_ordered() ->
19 ?FORALL(T,tree(),
20 ordered(to_list(T))).
21
22 ordered(Xs) ->
23 lists:usort(Xs) == Xs.
24
25 to_list(leaf) ->
26 [];
27 to_list({node,L,X,R}) ->
28 to_list(L) ++ [X] ++ to_list(R).
29

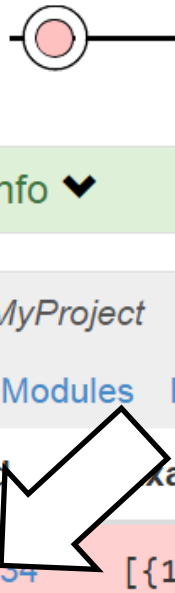

QuickCheck | Contin
quickcheck-ci.com/c
quickcheck-ci.com/cover/module/3450/trees.erl.html
1 -module(trees).
2 -include_lib("eqc/include/eqc.hrl").
3 -compile({parse_transform,eqc_cover}).
4 -compile(export_all).
5
6 tree() ->
7 ?SIZED(Size,tree(-Size,Size)).
8
9 tree(Lo,Hi) when Hi < Lo ->
10 leaf;
11 tree(Lo, Hi) ->
12 frequency([{1,leaf},
13 {2,?LET(X,choose(Lo,Hi),
14 ?LET({L,R},{tree(Lo,X-1),tree(X+1,Hi)},
15 ?SHRINK({node,L,X,R},
16 [leaf,L,R]))}]).
17
18 prop_ordered() ->
19 ?FORALL(T,tre
20 ordere
21
22 ordered(Xs) ->
23 lists:usort(Xs) == Xs.
24
25 to_list(leaf) ->
26 [];
27 to_list({node,L,X,R}) ->
28 to_list(L) ++ [X] ++ to_list(R).
trees:prop_ordered() : 1 call
quickcheck-ci.com/cover/module/3450/trees.erl.html#


QuickCheck | Contin
quickcheck-ci.com/c
quickcheck-ci.com/cover/module/3450/trees.erl.html
-module(trees).
-include_lib("eqc/include/eqc.hrl").
-compile({parse_transform,eqc_cover}).
-compile(export_all).
tree() ->
?SIZED(Size,tree(-Size,Size)).
tree(Lo,Hi) when Hi < Lo ->
leaf;
tree(Lo, Hi) ->
frequency([{1,leaf},
{2,?LET(X,choose(Lo,Hi),
?LET({L,R},{tree(Lo,X-1),tree(X+1,Hi)},
?SHRINK({node,L,X,R},
[leaf,L,R])))}]).
prop_ordered() ->
?FORALL(T,tree(),
ordered(to_list(T))).
ordered(Xs) ->
lists:
trees:prop_ordered() : 100 calls
to_list(leaf) ->
[];
to_list({node,L,X,R}) ->
to_list(L) ++ [X] ++ to_list(R).


QuickCheck | Contin
quickcheck-ci.com/c
quickcheck-ci.com/cover/module/3450/trees.erl.html
14 ?LET({L,R},{tree(Lo,X-1),tree(X+1,Hi)},
15 ?SHRINK({node,L,X,R},
16 [leaf,L,R]))}]).
17
18 prop_ordered() ->
19 ?FORALL(T,tree(),
20 ordered(to_list(T))).
21
22 ordered(Xs) ->
23 lists:usort(Xs.
24
25 to_list(leaf) ->
26 [];
27 to_list({node,L,X,R}) ->
28 to_list(L) ++
29
30 %% member
31
32 prop_member() ->
33 ?FORALL({X,T},{nat(),tree()},
34 equals(member(X,T),lists:member(X,to_list(T)))).
35
36 member(_,leaf) ->
37 false;
38 member(X,{node,L,Y,R}) ->
39 if
40 X<Y ->
41 member(X,R);
Total : 959 calls
trees:prop_ordered() : 524 calls
trees:prop_member() : 435 calls
quickcheck-ci.com/cover/module/3450/trees.erl.html#


QuickCheck | Contin
quickcheck-ci.com/c
quickcheck-ci.com
QuviQ
rjmh/trees build #1, revision 80 [71c0c9ad4a] - 25 Mar 2015 22:17
Queue build
Info
MyProject
Modules History Coverage
trees
Properties History Coverage
File
Coverage
trees.erl
100.0%


QuickCheck | Contin
quickcheck-ci.com/c
quickcheck-ci.com
QuviQ
rjmh/trees build #1, revision 80 [71c0c9ad4a] - 25 Mar 2015 22:17
Queue build
Info
MyProject
Modules History Coverage
trees
Properties History Coverage
Property
Output
Numtests
Runtime
prop_member
Shrinking (0 times) [+]
9
0.27s
prop_ordered
OK, passed 100 tests
100
0.12s



QuickCheck | Contin × quickcheck-ci.com/c ×

quickcheck-ci.com

QuviQ

rjmh/trees build #1, revision 80 [71c0c9ad4a] - 25 Mar 2015 22:17

Queue build

Info

MyProject Modules History Coverage

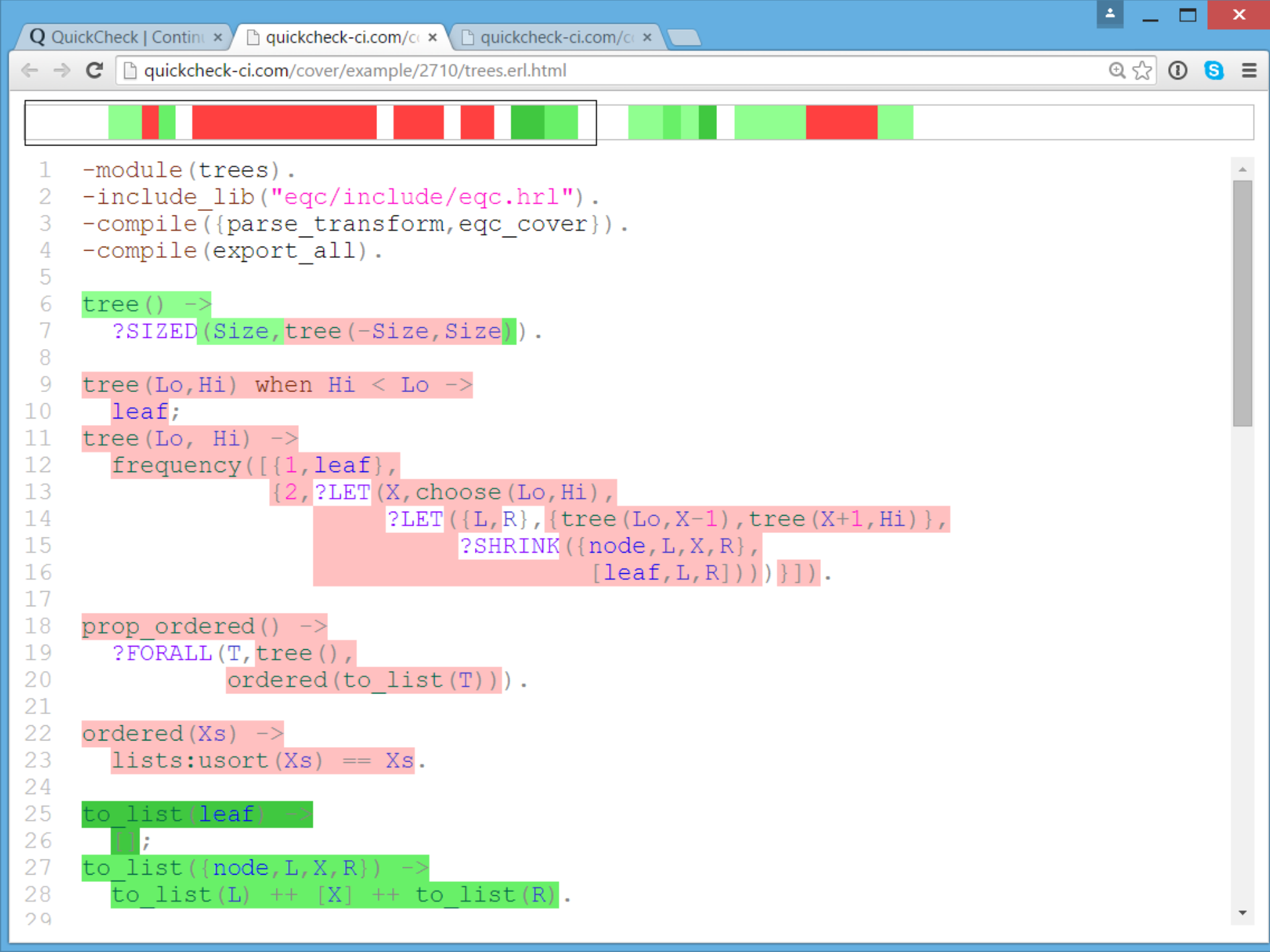
trees Properties History Coverage

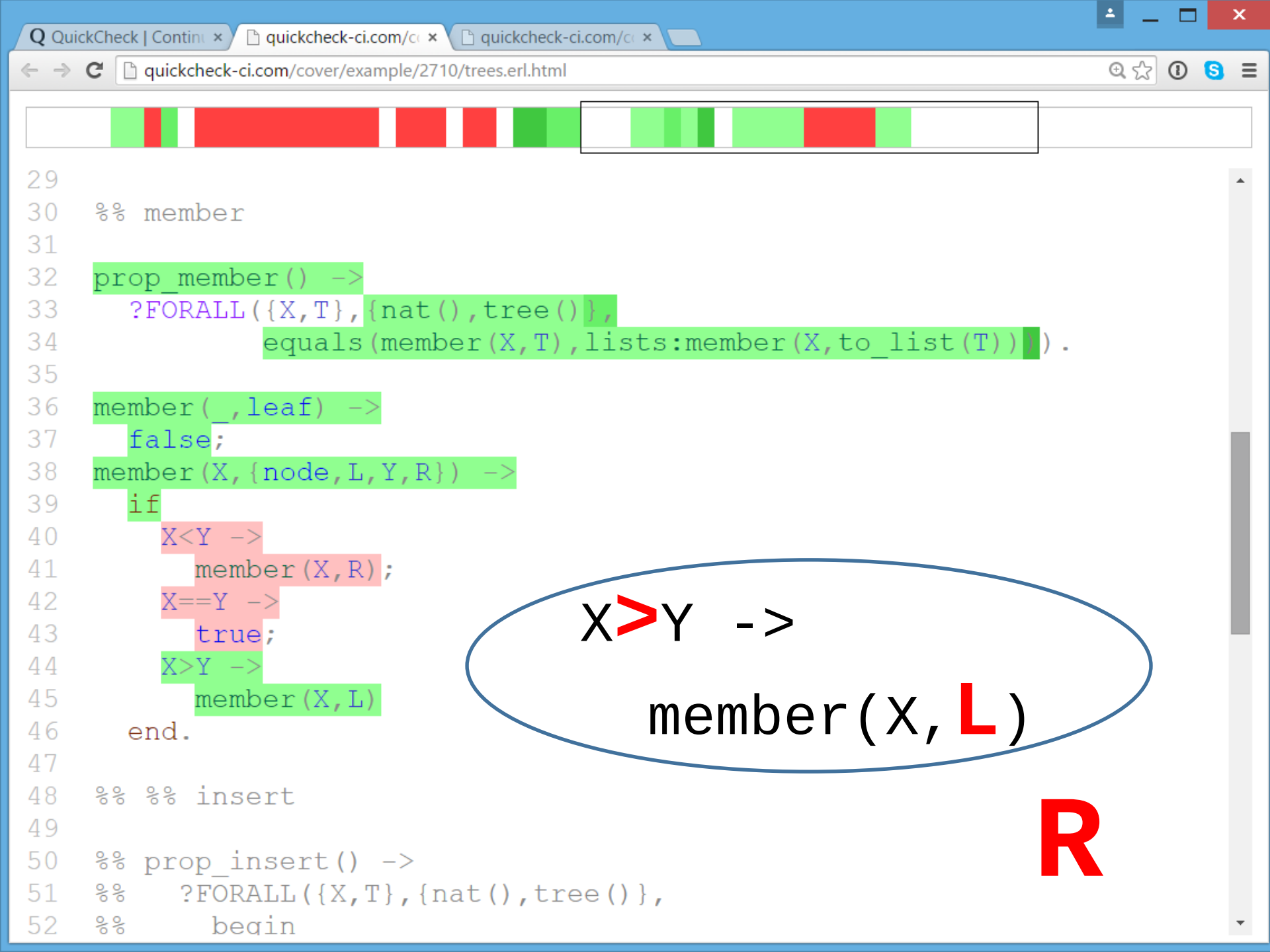
prop_member Examples History Coverage

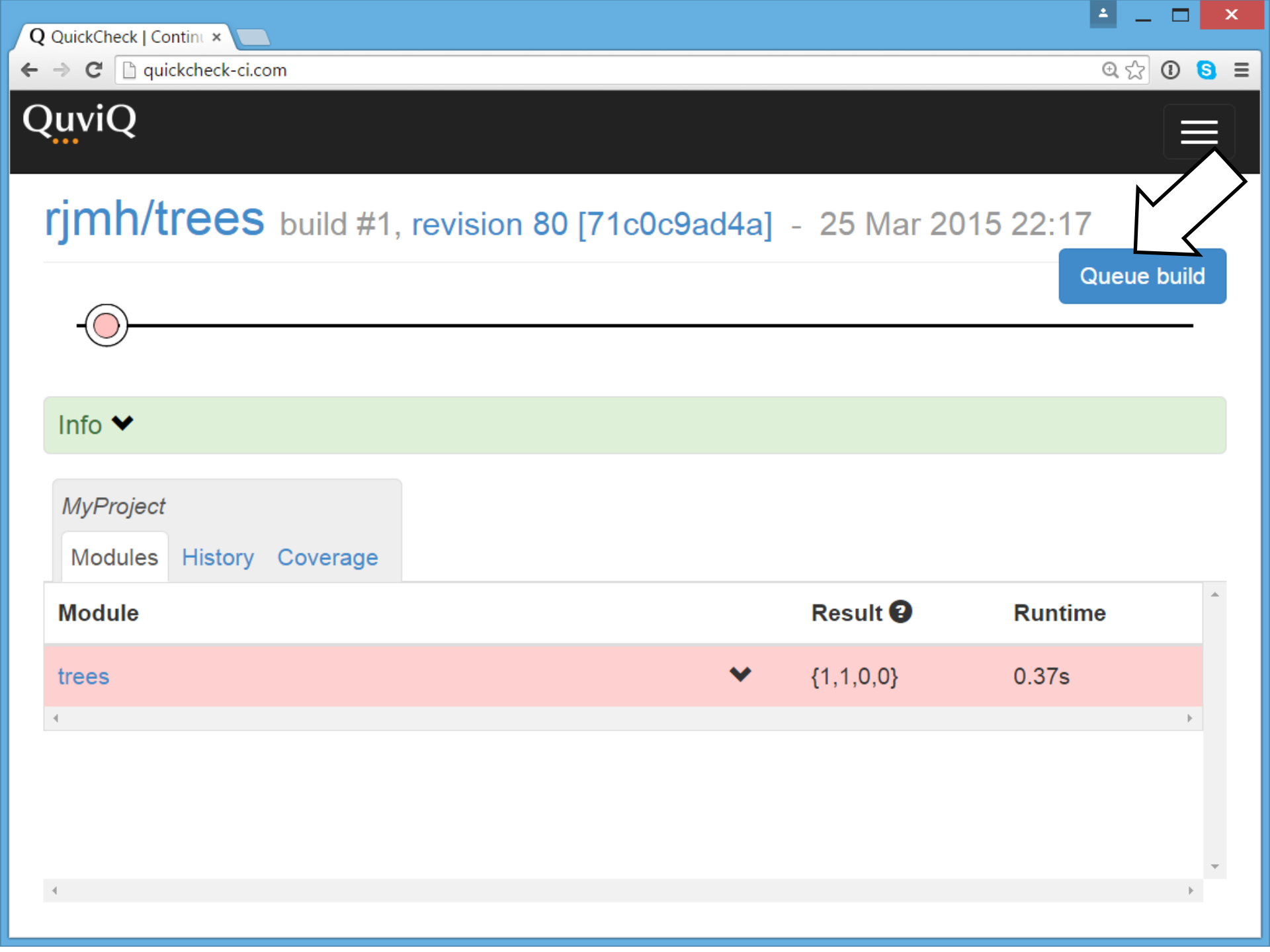
Id	Example	Output	Runtime	Active?
434	[{1,{node,leaf,-1,{node,leaf,1,leaf}}}]	Failed! [+]	0.00s	active
435	[{0,{node,{node,leaf,0,leaf},1,leaf}}]	Failed! [+]	0.00s	active
436	[{1,{node,leaf,0,{node,leaf,1,leaf}}}]	Failed! [+]	0.00s	active
437	[{0,{node,leaf,-1,{node,leaf,0,leaf}}}]	Failed! [+]	0.00s	active


QuickCheck | Contin
quickcheck-ci.com/c
quickcheck-ci.com
QuviQ
rjmh/trees build #1, revision 80 [71c0c9ad4a] - 25 Mar 2015 22:17
Queue build
Info
MyProject
Modules History Coverage
trees
Properties History Coverage
prop_member
Examples History Coverage
Example ID: 434
Details History Coverage
Failed!
{1,{node,leaf,-1,{node,leaf,1,leaf}}}
false /= true


QuickCheck | Contin
quickcheck-ci.com/c
quickcheck-ci.com
QuviQ
rjmh/trees build #1, revision 80 [71c0c9ad4a] - 25 Mar 2015 22:17
Queue build
Info
MyProject
Modules History Coverage
trees
Properties History Coverage
prop_member
Examples History Coverage
Example ID: 434
Details History Coverage
File
Coverage
trees.erl
44.4%
quickcheck-ci.com/#example_coverage


QuickCheck | Contin
quickcheck-ci.com/c
quickcheck-ci.com/c
quickcheck-ci.com/cover/example/2710/trees.erl.html
-module(trees).
-include_lib("eqc/include/eqc.hrl").
-compile({parse_transform,eqc_cover}).
-compile(export_all).
tree() ->
?SIZED(Size,tree(-Size,Size)).
tree(Lo,Hi) when Hi < Lo ->
leaf;
tree(Lo, Hi) ->
frequency([{1,leaf},
{2,?LET(X,choose(Lo,Hi),
?LET({L,R},{tree(Lo,X-1),tree(X+1,Hi)},
?SHRINK({node,L,X,R},
[leaf,L,R])))}]).
prop_ordered() ->
?FORALL(T,tree(),
ordered(to_list(T))).
ordered(Xs) ->
lists:usort(Xs) == Xs.
to_list(leaf) ->
[];
to_list({node,L,X,R}) ->
to_list(L) ++ [X] ++ to_list(R).


QuickCheck | Contin
quickcheck-ci.com/c
quickcheck-ci.com/c
quickcheck-ci.com/cover/example/2710/trees.erl.html
29
30 %% member
31
32 prop_member() ->
33 ?FORALL({X,T},{nat(),tree()},
34 equals(member(X,T),lists:member(X,to_list(T)))).
35
36 member(_,leaf) ->
37 false;
38 member(X,{node,L,Y,R}) ->
39 if
40 X<Y ->
41 member(X,R);
42 X==Y ->
43 true;
44 X>Y ->
45 member(X,L)
46 end.
47
48 %% %% insert
49
50 %% prop_insert() ->
51 %% ?FORALL({X,T},{nat(),tree()},
52 %% begin
X>Y ->
member(X,L)
R


QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/trees build #1, revision 80 [71c0c9ad4a] - 25 Mar 2015 22:17
Queue build
Info
MyProject
Modules
History
Coverage
Module
Result
Runtime
trees
{1,1,0,0}
0.37s

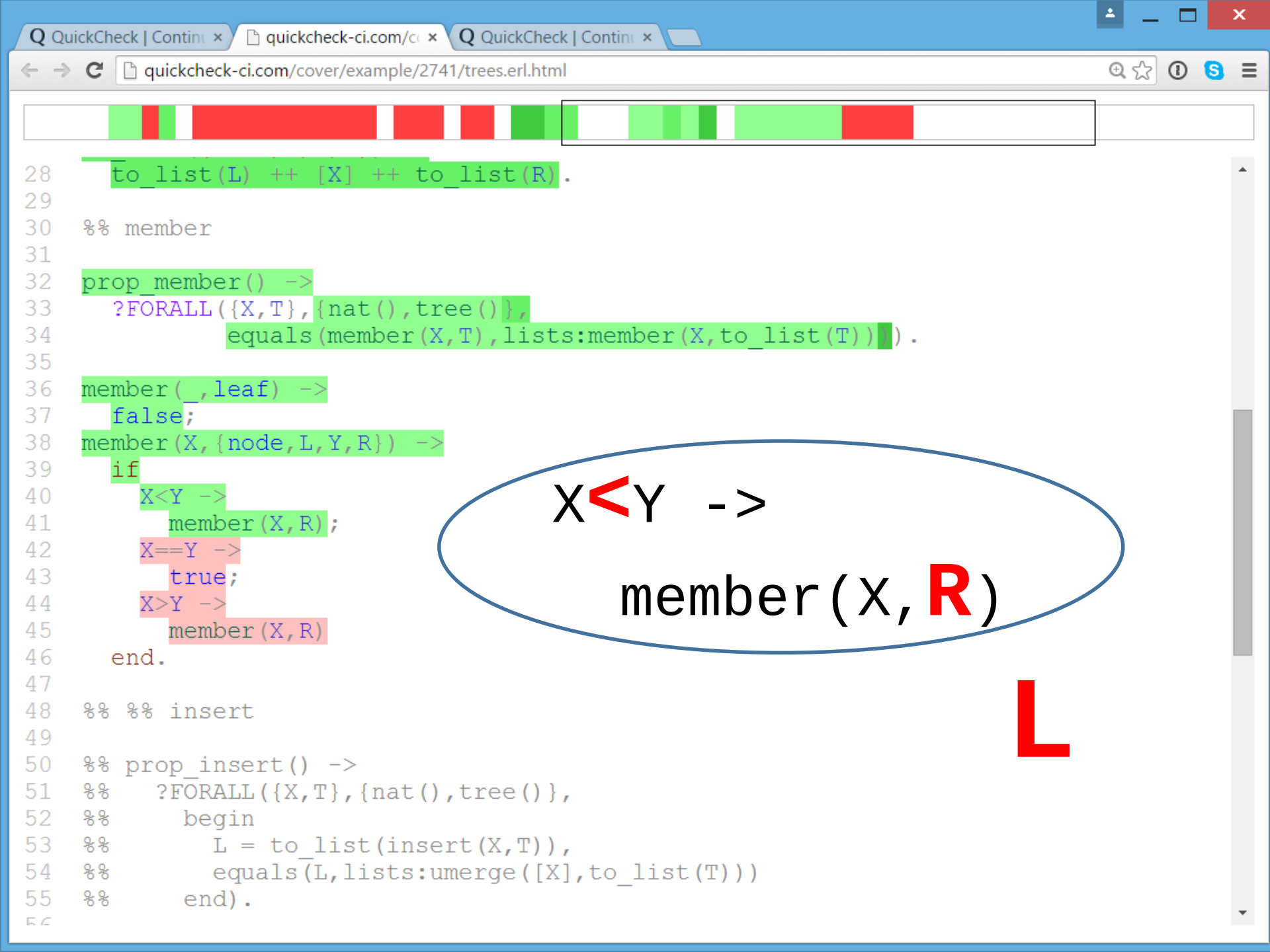

QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/trees build #2, revision 81 [b8d7ddff51] - 25 Mar 2015 23:14
Queue build
Info
MyProject
Modules
History
Coverage
Modul
Result
Runtime
trees
{1,1,0,0}
0.38s


QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/trees build #2, revision 81 [b8d7ddff51] - 25 Mar 2015 23:14
Queue build
Info
MyProject
Modules History Coverage
trees
Properties History Coverage
Output
Numtests
Runtime
prop_member
Shrinking (2 times) [+]
14
0.31s
prop_ordered
OK, passed 100 tests
100
0.12s


QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/trees build #2, revision 81 [b8d7ddff51] - 25 Mar 2015 23:14
Queue build
Info
MyProject
Modules History Coverage
trees
Properties History Coverage
prop_member
Examples History Coverage
Id Example Output Runtime Activ
458 [{0,{node,leaf,-1,{node,leaf,0,leaf}}}] OK, passed the test. [+] 0.00s active
455 [{0,{node,{node,leaf,0,leaf},1,leaf}}] Failed! [+] 0.03s active
454 [{3,{node,leaf,0,{node,leaf,3,leaf}}}] OK, passed the test. [+] 0.00s active
456 [{1,{node,leaf,0,{node,leaf,1,leaf}}}] OK, passed the test. [+] 0.00s active


QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/trees build #2, revision 81 [b8d7ddff51] - 25 Mar 2015 23:14
Queue build
Info
MyProject
Modules History Coverage
trees
Properties History Coverage
prop_member
Examples History Coverage
Example ID: 453
Details History Coverage
Failed!
{0,{node,{node,leaf,0,leaf},1,leaf}}
false /= true

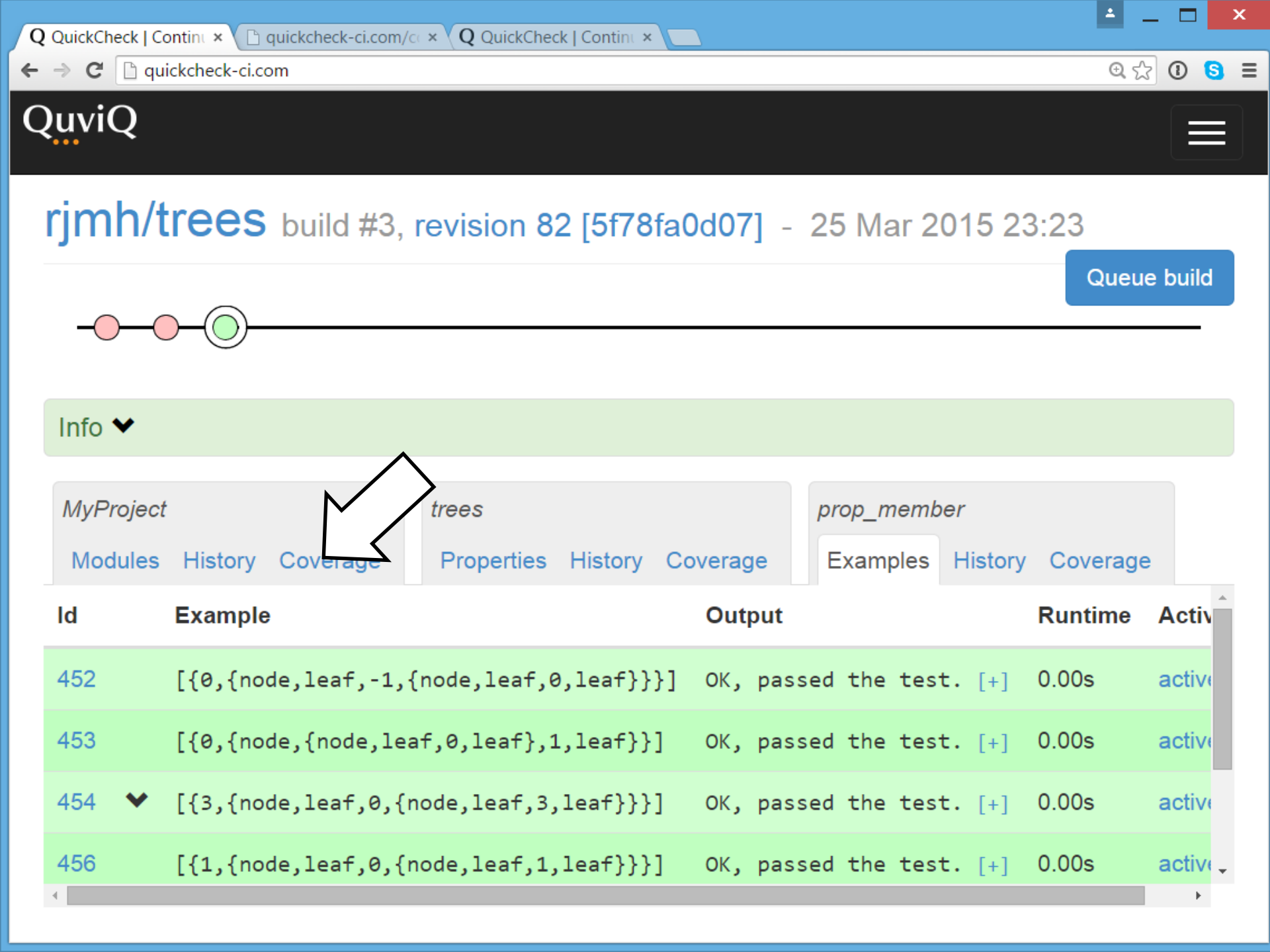

QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/trees build #2, revision 81 [b8d7ddff51] - 25 Mar 2015 23:14
Queue build
Info
MyProject
Modules History Coverage
trees
Properties History Coverage
prop_member
Examples History Coverage
Example ID: 453
Details History Coverage
File
Coverage
trees.erl
44.4%
quickcheck-ci.com/#example_coverage

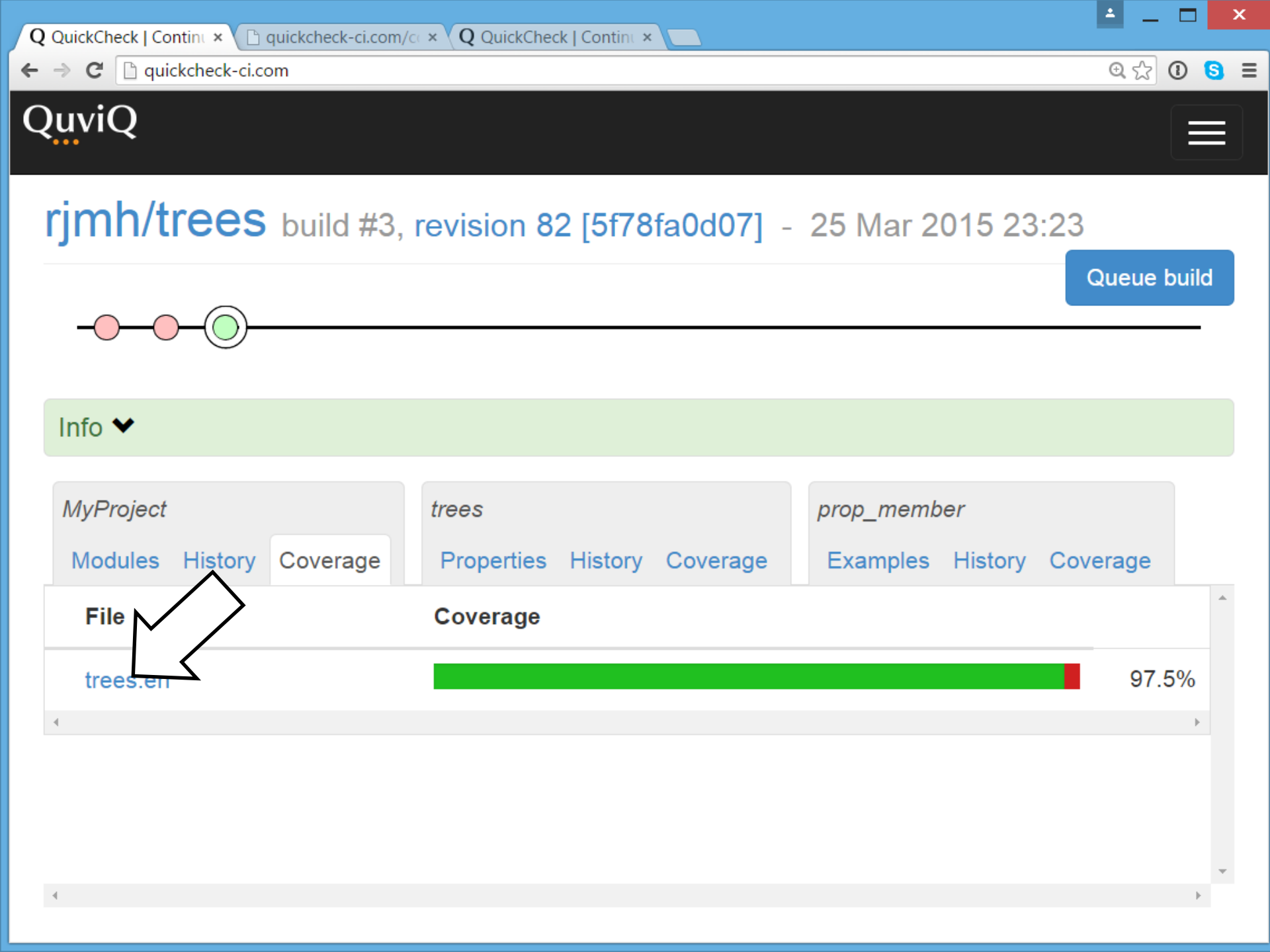

QuickCheck | Contin
quickcheck-ci.com/c
QuickCheck | Contin
quickcheck-ci.com/cover/example/2741/trees.erl.html
to_list(L) ++ [X] ++ to_list(R).
%% member
prop_member() ->
?FORALL({X,T},{nat(),tree()},
equals(member(X,T),lists:member(X,to_list(T)))).
member(_,leaf) ->
false;
member(X,{node,L,Y,R}) ->
if
X<Y ->
member(X,R);
X==Y ->
true;
X>Y ->
member(X,R)
end.
%% %% insert
%% prop_insert() ->
%% ?FORALL({X,T},{nat(),tree()},
%% begin
%% L = to_list(insert(X,T)),
%% equals(L,lists:umerge([X],to_list(T)))
%% end).
X<Y ->
member(X,R)
L


QuickCheck | Contin
quickcheck-ci.com/c
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/trees build #2, revision 81 [b8d7ddff51] - 25 Mar 2015 23:14
Queue build
Info
MyProject
Modules History Coverage
trees
Properties History Coverage
prop_member
Examples History Coverage
Example ID: 453
Details History Coverage
File
Coverage
trees.erl
44.4%


QuickCheck | Contin
quickcheck-ci.com/c
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/trees build #3, revision 82 [5f78fa0d07] - 25 Mar 2015 23:23
Queue build
Info
MyProject
Modules History Coverage
trees
Properties History Coverage
prop_member
Examples History Coverage
Result
Runtime
{2,0,0,0}
0.40s


QuickCheck | Contin
quickcheck-ci.com/c
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/trees build #3, revision 82 [5f78fa0d07] - 25 Mar 2015 23:23
Queue build
Info
MyProject
Modules History Coverage
trees
Properties History Coverage
prop_member
Examples History Coverage
Property
Output
Numtests
Runtime
prop_me
OK, passed 100 tests
107
0.13s
prop_ordered
OK, passed 100 tests
100
0.27s


QuickCheck | Contin
quickcheck-ci.com/c
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/trees build #3, revision 82 [5f78fa0d07] - 25 Mar 2015 23:23
Queue build
Info
MyProject
Modules History Coverage
trees
Properties History Coverage
prop_member
Examples History Coverage
Id Example Output Runtime Activ
452 [{0,{node,leaf,-1,{node,leaf,0,leaf}}}] OK, passed the test. [+] 0.00s
453 [{0,{node,{node,leaf,0,leaf},1,leaf}}] OK, passed the test. [+] 0.00s
454 [{3,{node,leaf,0,{node,leaf,3,leaf}}}] OK, passed the test. [+] 0.00s
456 [{1,{node,leaf,0,{node,leaf,1,leaf}}}] OK, passed the test. [+] 0.00s


QuickCheck | Contin
quickcheck-ci.com/c
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/trees build #3, revision 82 [5f78fa0d07] - 25 Mar 2015 23:23
Queue build
Info
MyProject
Modules History Coverage
trees
Properties History Coverage
prop_member
Examples History Coverage
File
Coverage
trees.erl
97.5%


QuickCheck | Contin
quickcheck-ci.com/c
quickcheck-ci.com/c
QuickCheck | Contin
quickcheck-ci.com/cover/build/771/trees.erl.html
1 -module(trees).
2 -include_lib("eqc/include/eqc.hrl").
3 -compile({parse_transform,eqc_cover}).
4 -compile(export_all).
5
6 tree() ->
7 ?SIZED(Size,tree(-Size,Size)).
8
9 tree(Lo,Hi) when Hi < Lo ->
10 leaf;
11 tree(Lo, Hi) ->
12 frequency([{1,leaf},
13 {2,?LET(X,choose(Lo,Hi),
14 ?LET({L,R},{tree(Lo,X-1),tree(X+1,Hi)},
15 ?SHRINK({node,L,X,R},
16 [leaf,L,R]))}]).
17
18 prop_ordered() ->
19 ?FORALL(T,tree(),
20 ordered(to_list(T))).
21
22 ordered(Xs) ->
23 lists:usort(Xs) == Xs.
24
25 to_list(leaf) ->
26 [];
27 to_list({node,L,X,R}) ->
28 to_list(L) ++ [X] ++ to_list(R).
29
Code which is only used to *shrink failing tests*

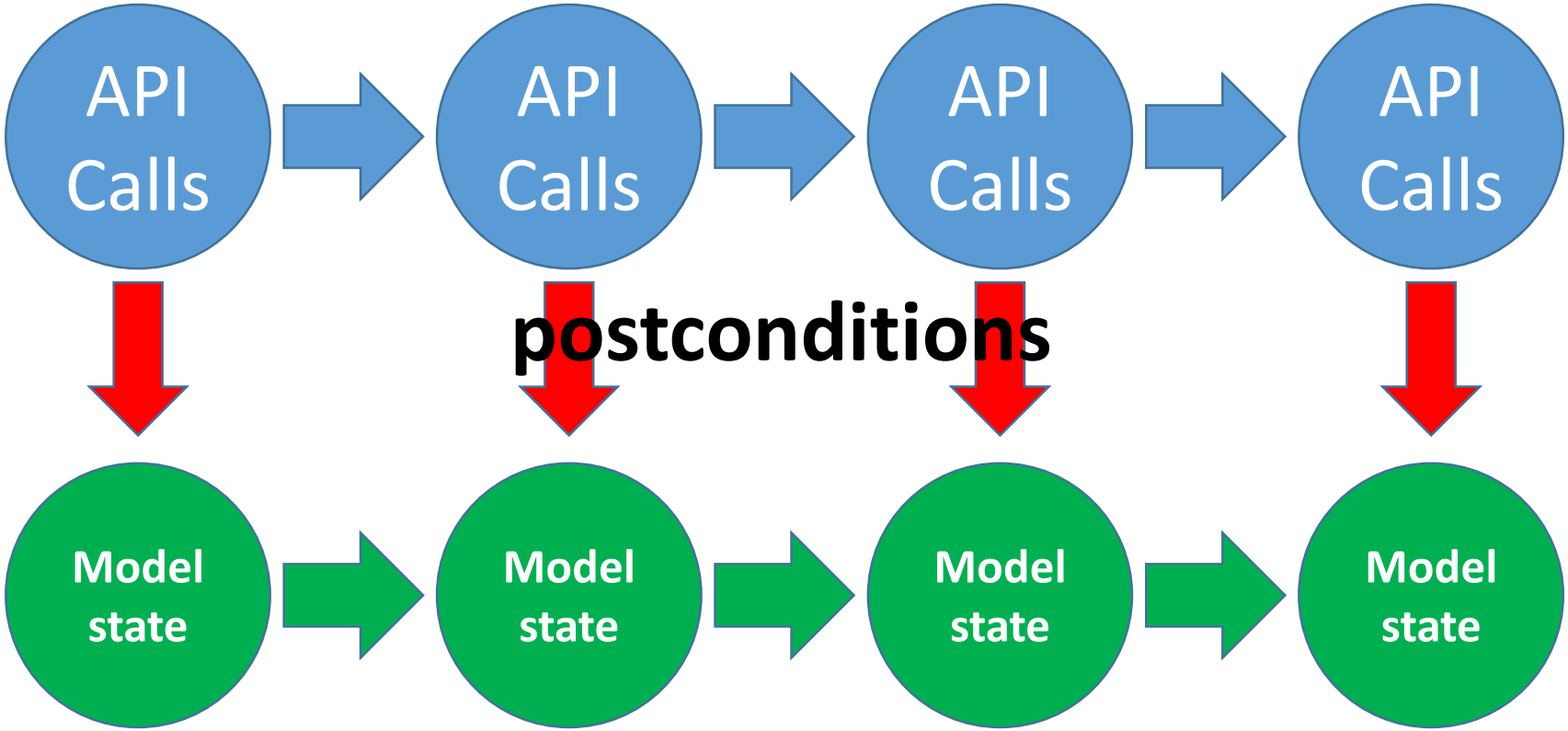
QuickCheck CI

- Builds a regression test suite automatically
  - See progress in terms of tests which now pass
  - Save *rare* tests which were hard to find
- Presents coverage information in depth
  - See at a glance what has been tested
  - See the effects of test case *distribution*
  - Helps localize bugs!

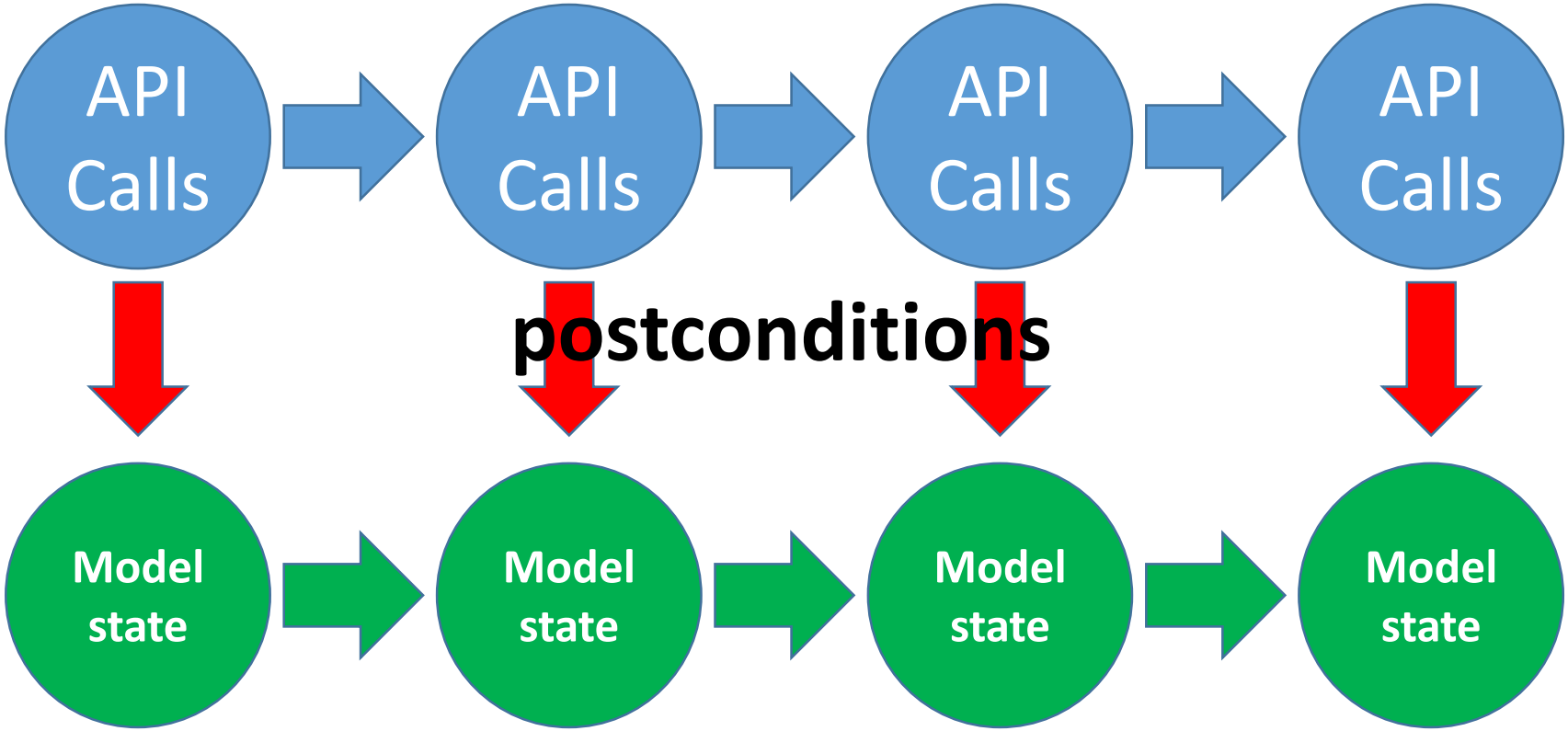
State machine testing—example

- Let's test the process registry
  - register(Name,Pid)
  - unregister(Name)
  - spawn()—to create pids for test data
- What's different now?
  - These functions change the state of the registry
- Not looking for bugs!
  - We'll reverse engineer *preconditions* instead

State Machine Models



State Machine Models



Modelling the registry state

```
#state{
   pids = […<0.32.0>…],
   regs = [{a,<0.32.0>},…]
}
```

Added by spawn

Added by register,
removed by unregister

Specification of register

```
register_pre(S) ->
  S#state.pids /= [].

register_args(S) ->
  [name(),elements(S#state.pids)].

register(Name,Pid) ->
  erlang:register(Name,Pid).

register_next(S,_,[Name,Pid]) ->
  S#state{regs=S#state.regs++[{Name,Pid}]}.
```

Isn’t this just what register does?

```
int erts_register_name(Process *c_p, Eterm name, Eterm id)
{
    int res = 0;
    Process *proc = NULL;
    Port *port = NULL;
    RegProc r, *rp;
    ERTS_SMP_CHK_HAVE_ONLY_MAIN_PROC_LOCK(c_p);

    if (is_not_atom(name) || name == am_undefined)
        return res;

    if (c_p->common.id == id) /* A very common case I think... */
        proc = c_p;
    else {
        if (is_not_internal_pid(id) && is_not_internal_port(id))
            return res;
        erts_smp_proc_unlock(c_p, ERTS_PROC_LOCK_MAIN);
        if (is_internal_port(id)) {
            port = erts_id2port(id);
            if (!port)
                goto done;
```

Isn’t this just what register does?

```
        }
    }

#ifdef ERTS_SMP
    {
        ErtsProcLocks proc_locks = proc ? ERTS_PROC_LOCK_MAIN :
0;
        reg_safe_write_lock(proc, &proc_locks);

        if (proc && !proc_locks)
            erts_smp_proc_lock(c_p, ERTS_PROC_LOCK_MAIN);
    }
#endif

    if (is_internal_pid(id)) {
        if (!proc)
            proc = erts_pid2proc(NULL, 0, id,
ERTS_PROC_LOCK_MAIN);
        r.p = proc;
        if (!proc)
            goto done;
```

Isn’t this just what register does?

```
    if (proc->common.u.alive.reg)
        goto done;
    r.pt = NULL;
}
else {
    ASSERT(!INVALID_PORT(port, id));
    ERTS_SMP_LC_ASSERT(erts_lc_is_port_locked(port));
    r.pt = port;
    if (r.pt->common.u.alive.reg)
        goto done;
    r.p = NULL;
}

r.name = name;

rp = (RegProc*) hash_put(&process_reg, (void*) &r);
if (proc && rp->p == proc) {
    if (IS_TRACED_FL(proc, F_TRACE_PROCS)) {
        trace_proc(c_p, proc, am_register, name);
    }
```

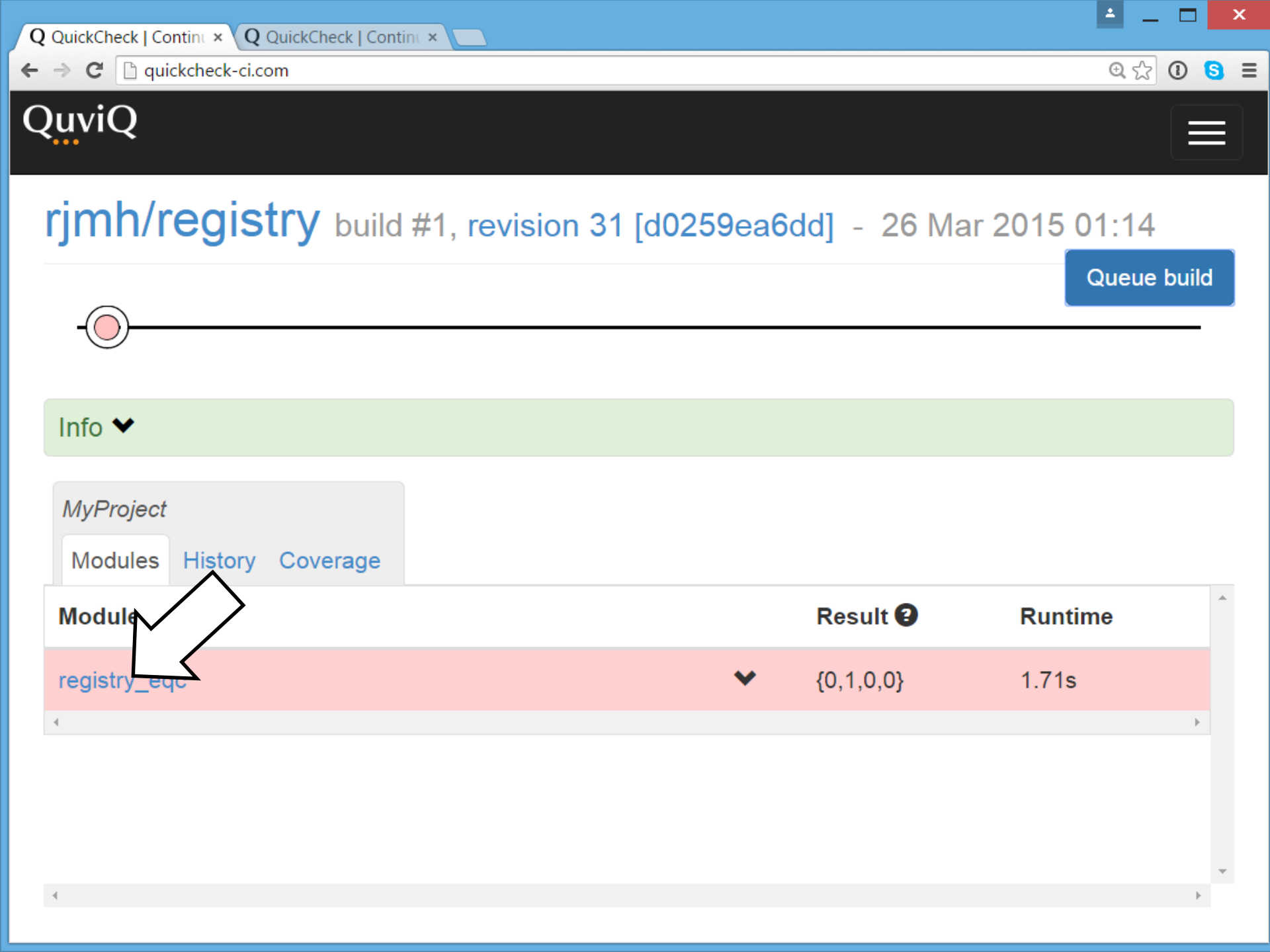
Isn’t this just what register does?

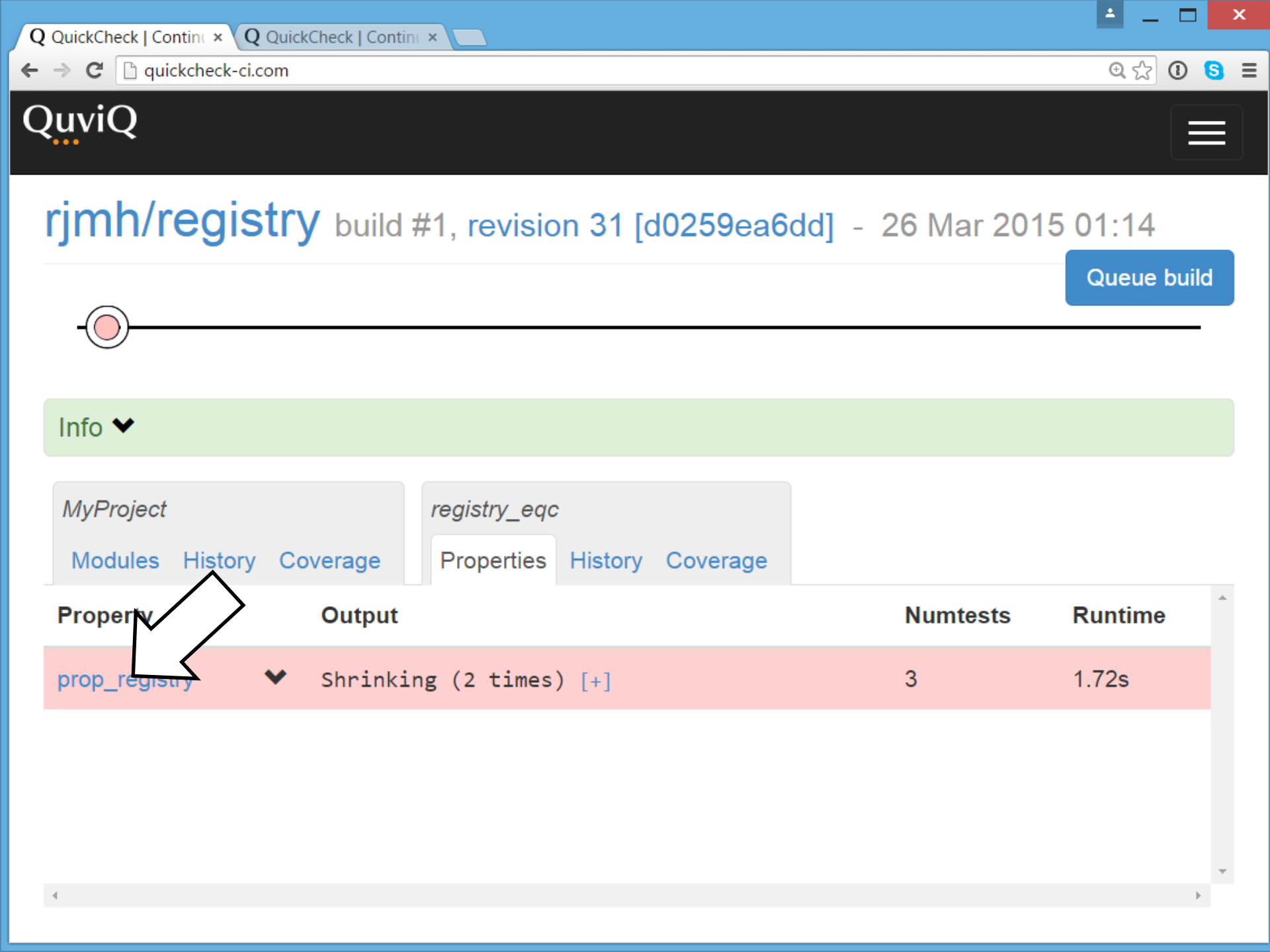
```
        proc->common.u.alive.reg = rp;
    }
    else if (port && rp->pt == port) {
        if (IS_TRACED_FL(port, F_TRACE_PORTS)) {
                trace_port(port, am_register, name);
        }
        port->common.u.alive.reg = rp;
    }

    if ((rp->p && rp->p->common.id == id)
        || (rp->pt && rp->pt->common.id == id)) {
        res = 1;
    }

done:
    reg_write_unlock();
    if (port)
        erts_port_release(port);
    if (c_p != proc) {
        if (proc)
            erts_smp_proc_unlock(proc, ERTS_PROC_LOCK_MAIN);
```

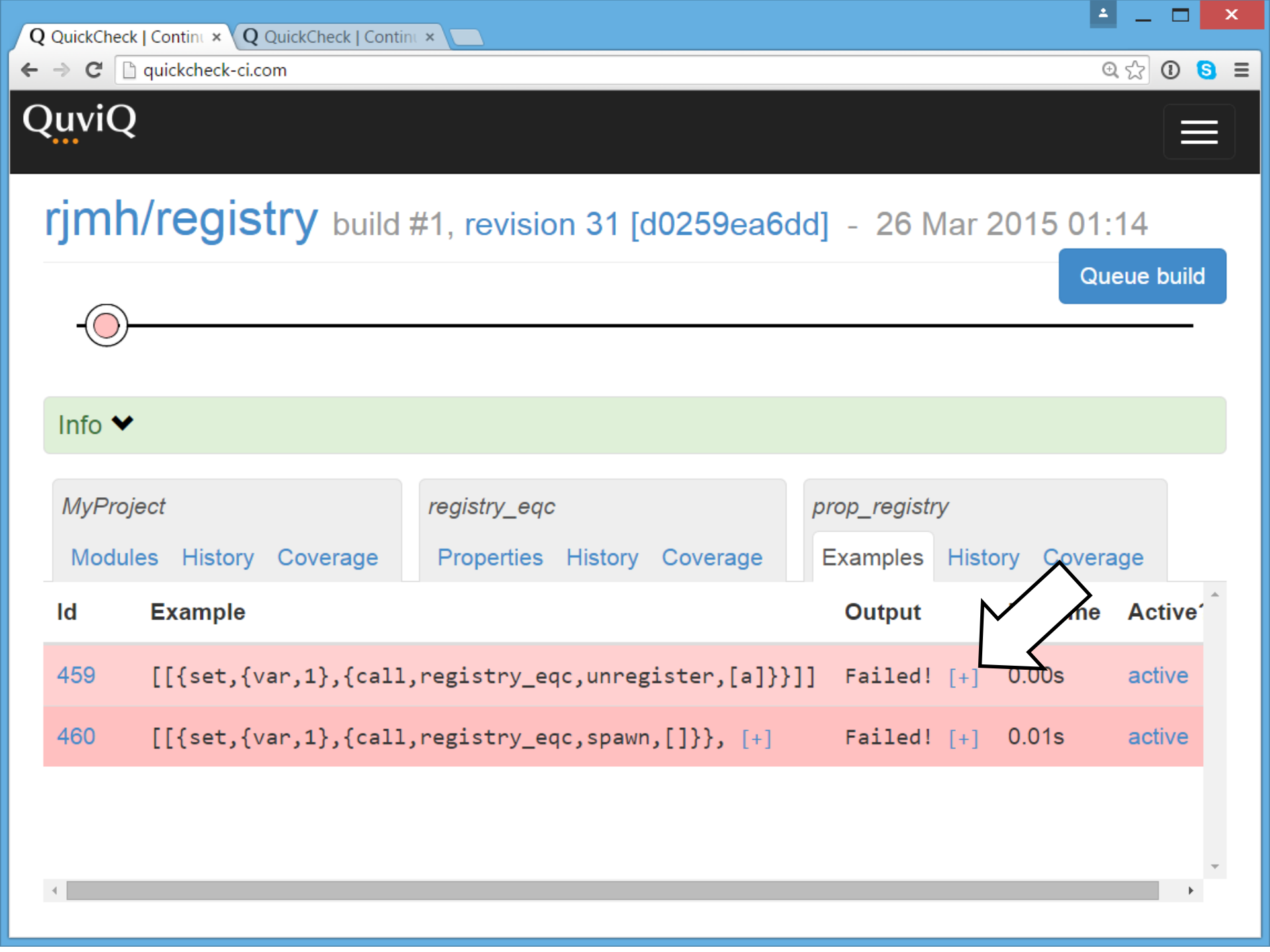
Isn’t this just what register does?

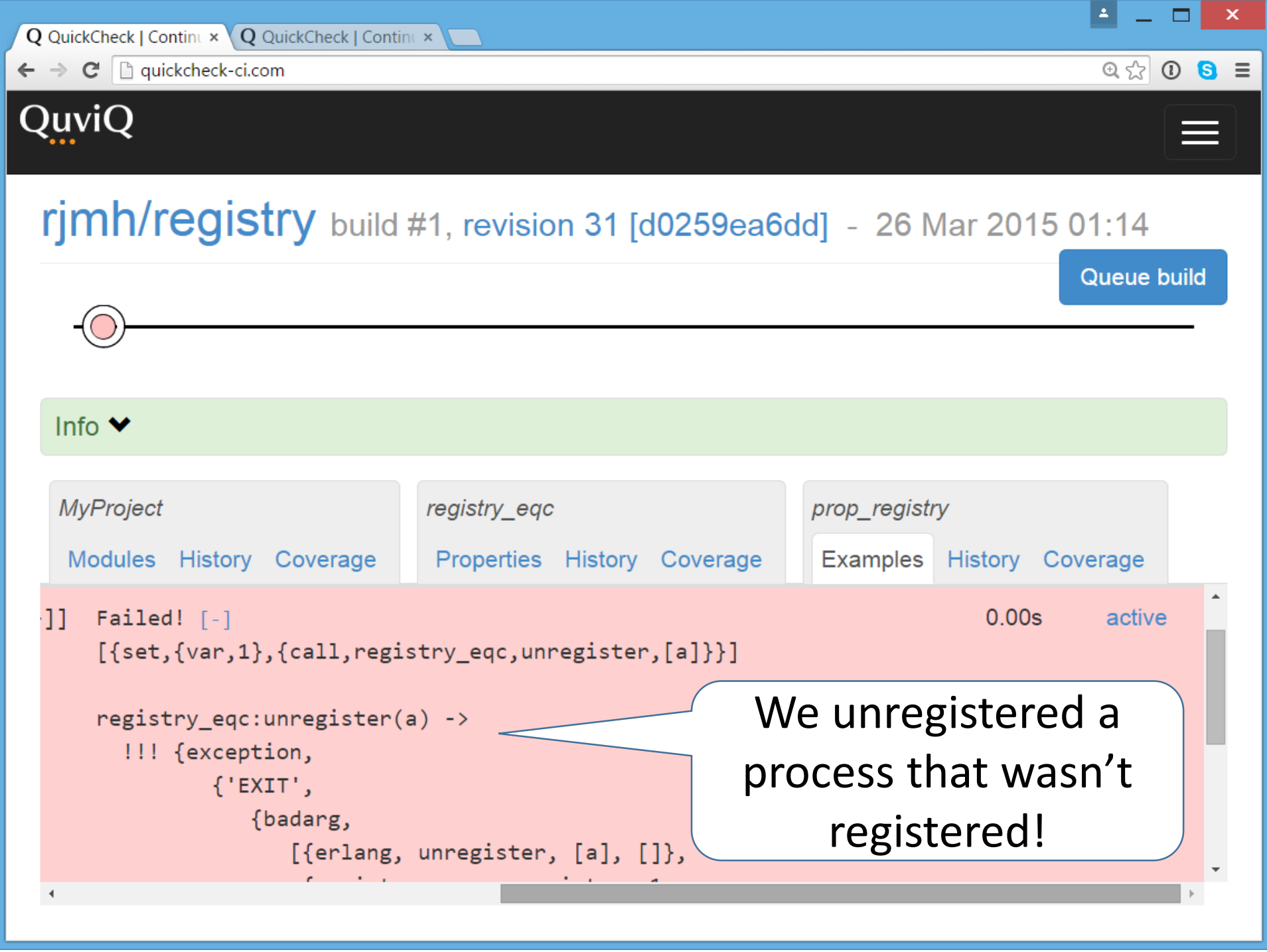
```
        erts_smp_proc_lock(c_p, ERTS_PROC_LOCK_MAIN);
    }
    return res;
}
```

DEMO


QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/registry build #1, revision 31 [d0259ea6dd] - 26 Mar 2015 01:14
Queue build
Info
MyProject
Modules
History
Coverage
Module
Result
Runtime
registry_eqc
{0,1,0,0}
1.71s


QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/registry build #1, revision 31 [d0259ea6dd] - 26 Mar 2015 01:14
Queue build
Info
MyProject
Modules History Coverage
registry_eqc
Properties History Coverage
Property
Output
Numtests
Runtime
prop_registry
Shrinking (2 times) [+]
3
1.72s


QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/registry build #1, revision 31 [d0259ea6dd] - 26 Mar 2015 01:14
Queue build
Info
MyProject
Modules History Coverage
registry_eqc
Properties History Coverage
prop_registry
Examples History Coverage
Id Example Output me Active
459 [[{set,{var,1},{call,registry_eqc,unregister,[a]}}]] Failed! [+] 0.00s active
460 [[{set,{var,1},{call,registry_eqc,spawn,[]}}, [+] Failed! [+] 0.01s active

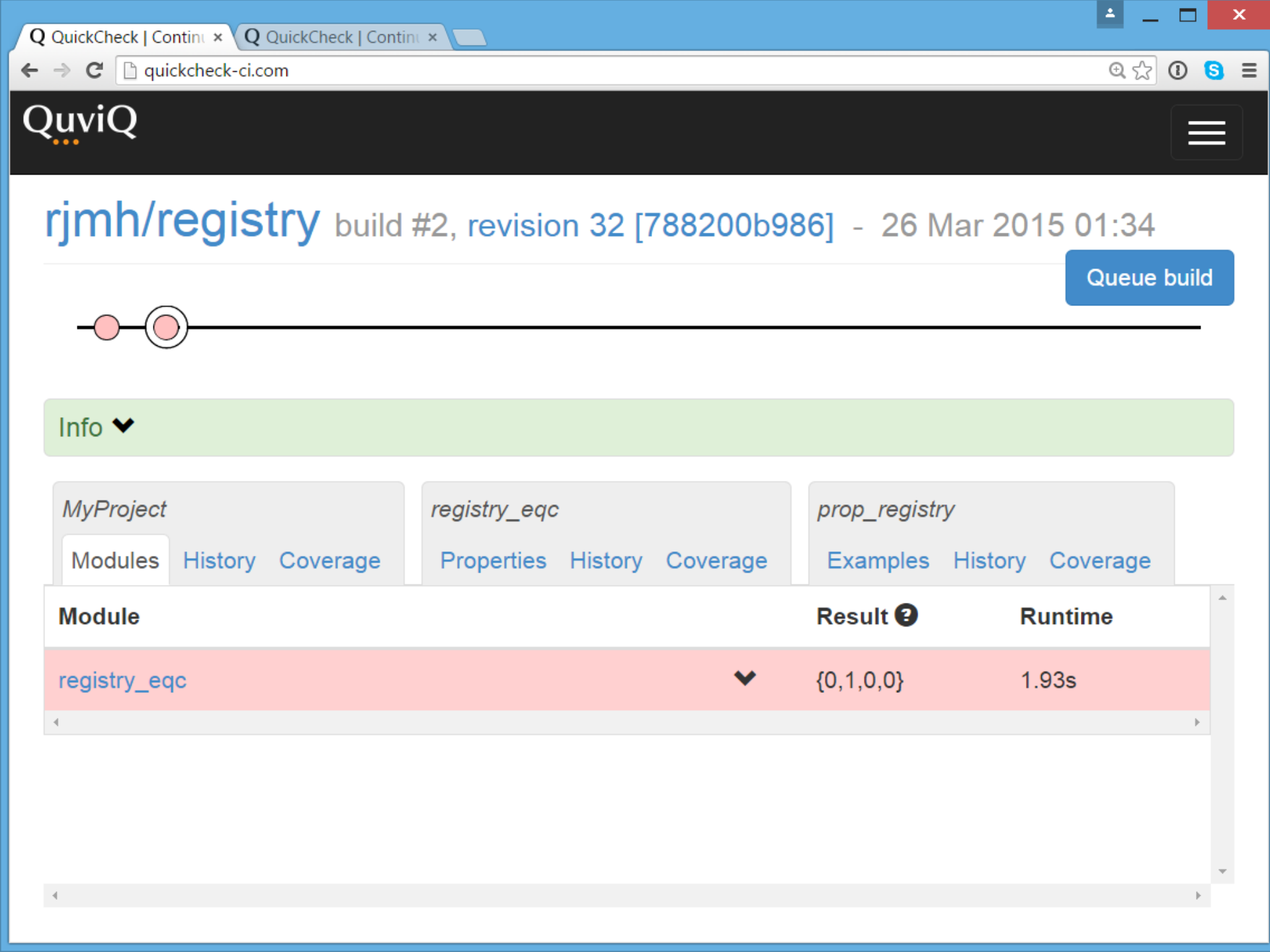

QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/registry build #1, revision 31 [d0259ea6dd] - 26 Mar 2015 01:14
Queue build
Info
MyProject
Modules History Coverage
registry_eqc
Properties History Coverage
prop_registry
Examples History Coverage
]] Failed! [-] 0.00s active
[{set,{var,1},{call,registry_eqc,unregister,[a]}}]
registry_eqc:unregister(a) ->
!!! {exception,
{'EXIT',
{badarg,
[{erlang, unregister, [a], []},
We unregistered a process that wasn't registered!

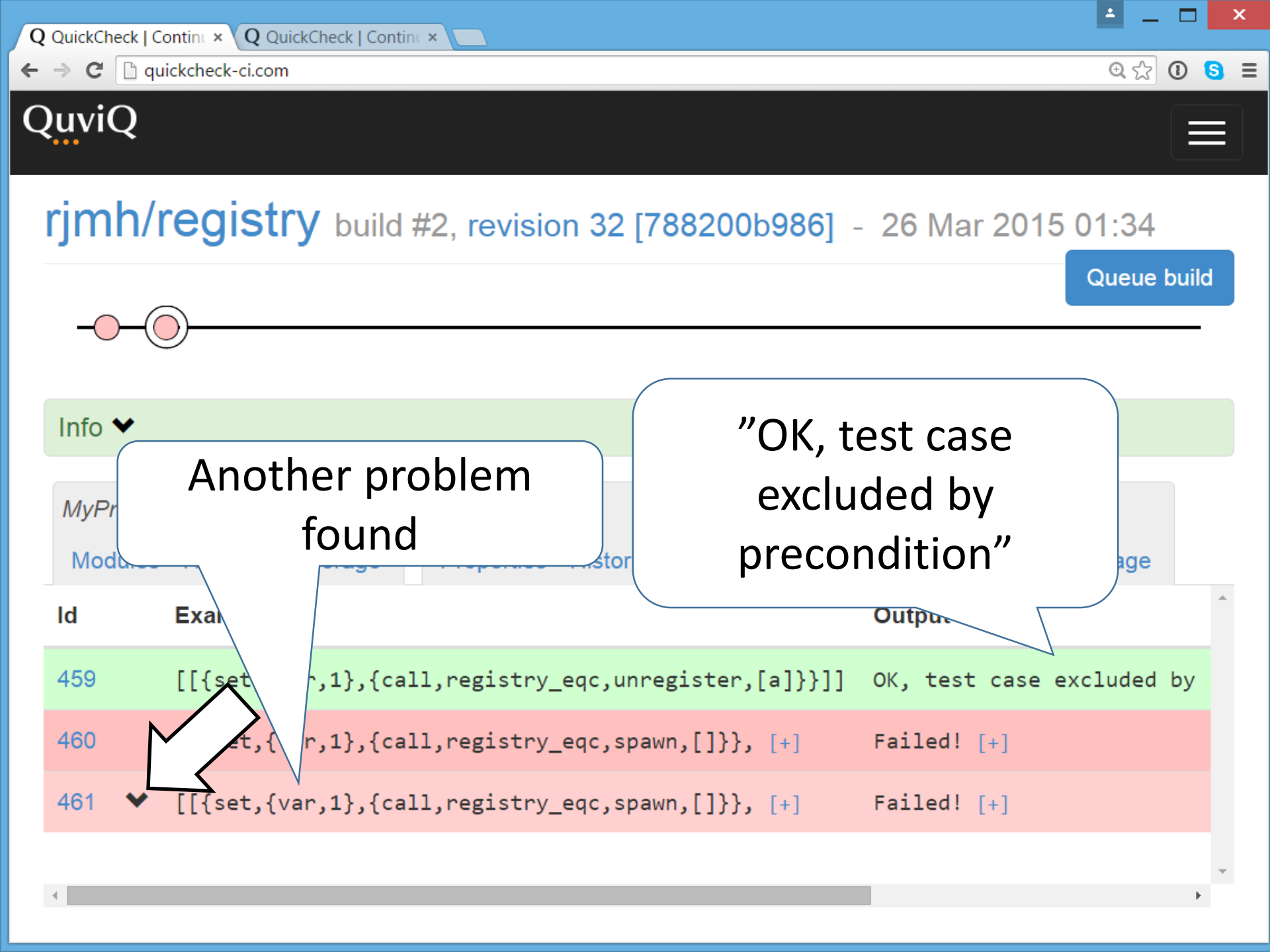

QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/registry build #1, revision 31 [d0259ea6dd] - 26 Mar 2015 01:14
Queue build
Info
MyProject
Modules History Coverage
registry_eqc
Properties History Coverage
prop_registry
Examples History Coverage
registry_eqc:spawn() -> <0.2458.0>
registry_eqc:register(a, <0.2458.0>) -> true
registry_eqc:register(a, <0.2458.0>) ->
!!! {exception,
{'EXIT',
{badarg,
[{erlang, register, [a, <0.2458.0>], []},
{registry_eqc, register, 2,
We registered the same name twice

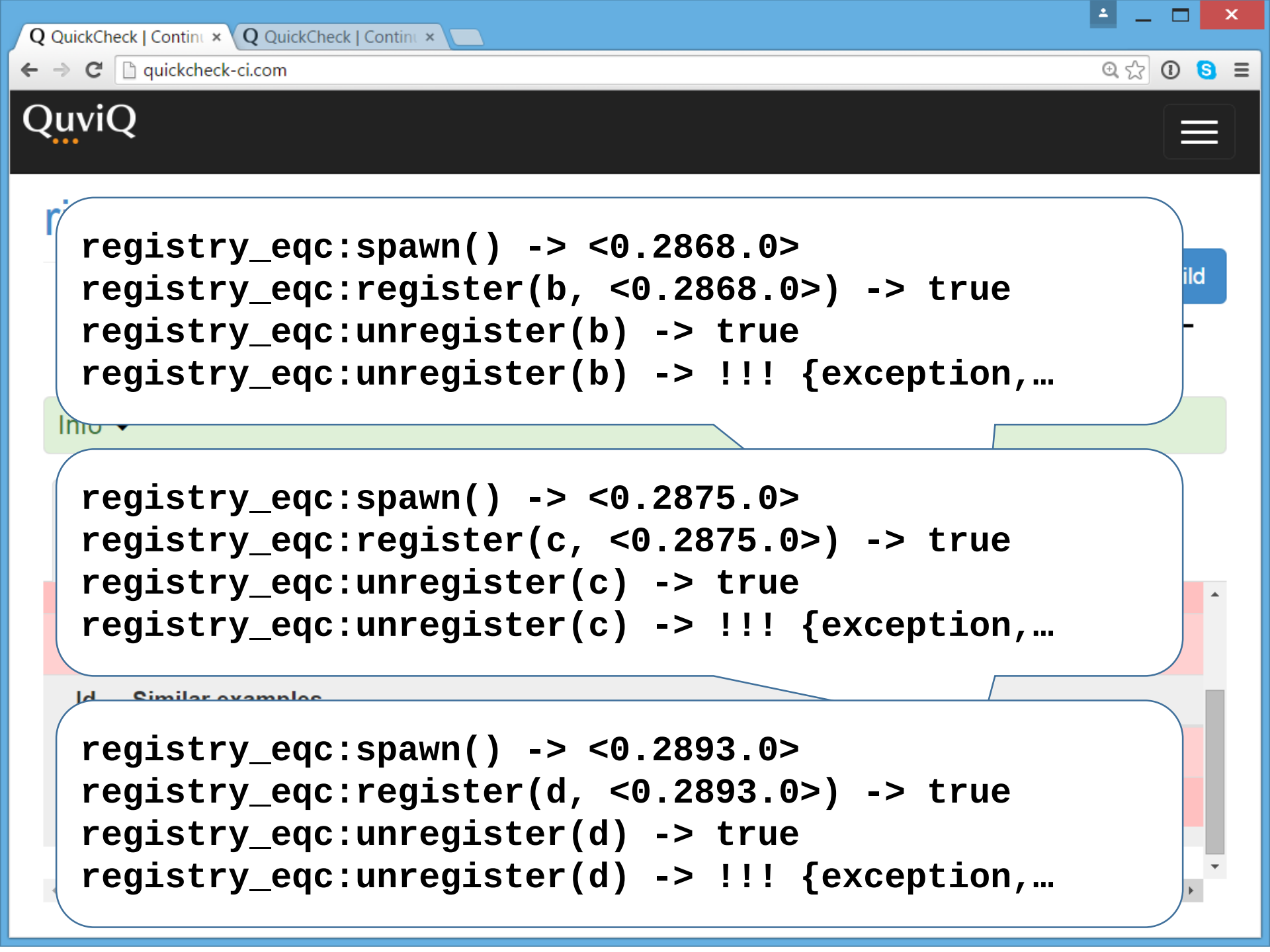
The specification of unregister

```
%% unregister

unregister_args(_) ->
  [name()].

unregister(Name) ->
  erlang:unregister(Name).

unregister_pre(S,[Name]) ->
  lists:keymember(Name,1,S#state.regs).
```

DON'T CALL unregister on a name not registered

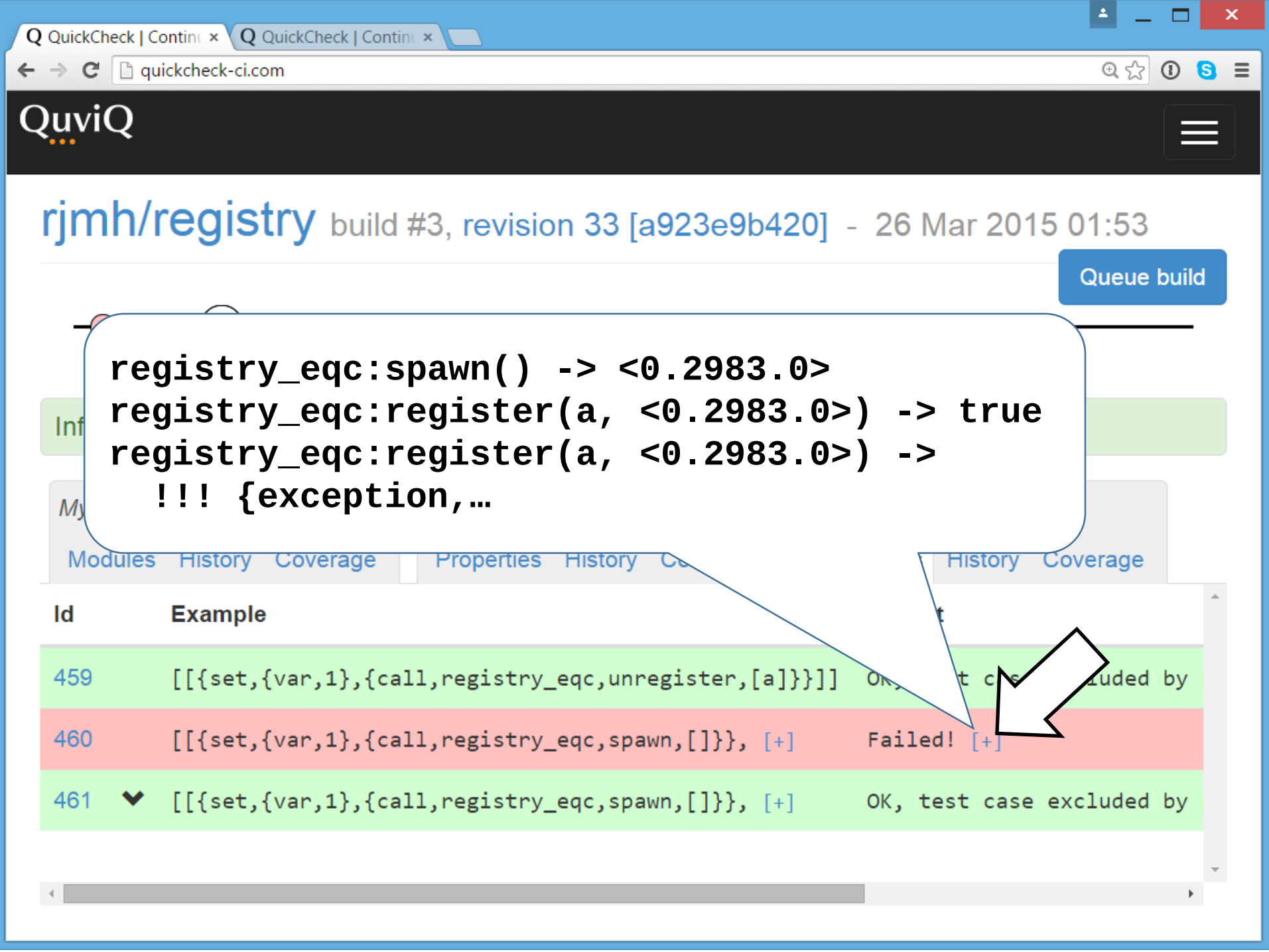

QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/registry build #2, revision 32 [788200b986] - 26 Mar 2015 01:34
Queue build
Info
MyProject
Modules History Coverage
registry_eqc
Properties History Coverage
prop_registry
Examples History Coverage
Module
Result
Runtime
registry_eqc
{0,1,0,0}
1.93s


QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/registry build #2, revision 32 [788200b986] - 26 Mar 2015 01:34
Queue build
Info
Another problem found
”OK, test case excluded by precondition”
Id
459
460
461
[[{set,{var,1},{call,registry_eqc,unregister,[a]}}]]
OK, test case excluded by
[[{set,{var,1},{call,registry_eqc,spawn,[]}}, [+]
Failed! [+]
[[{set,{var,1},{call,registry_eqc,spawn,[]}}, [+]
Failed! [+]


QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
registry_eqc:spawn() -> <0.2868.0>
registry_eqc:register(b, <0.2868.0>) -> true
registry_eqc:unregister(b) -> true
registry_eqc:unregister(b) -> !!! {exception,…
registry_eqc:spawn() -> <0.2875.0>
registry_eqc:register(c, <0.2875.0>) -> true
registry_eqc:unregister(c) -> true
registry_eqc:unregister(c) -> !!! {exception,…
registry_eqc:spawn() -> <0.2893.0>
registry_eqc:register(d, <0.2893.0>) -> true
registry_eqc:unregister(d) -> true
registry_eqc:unregister(d) -> !!! {exception,…

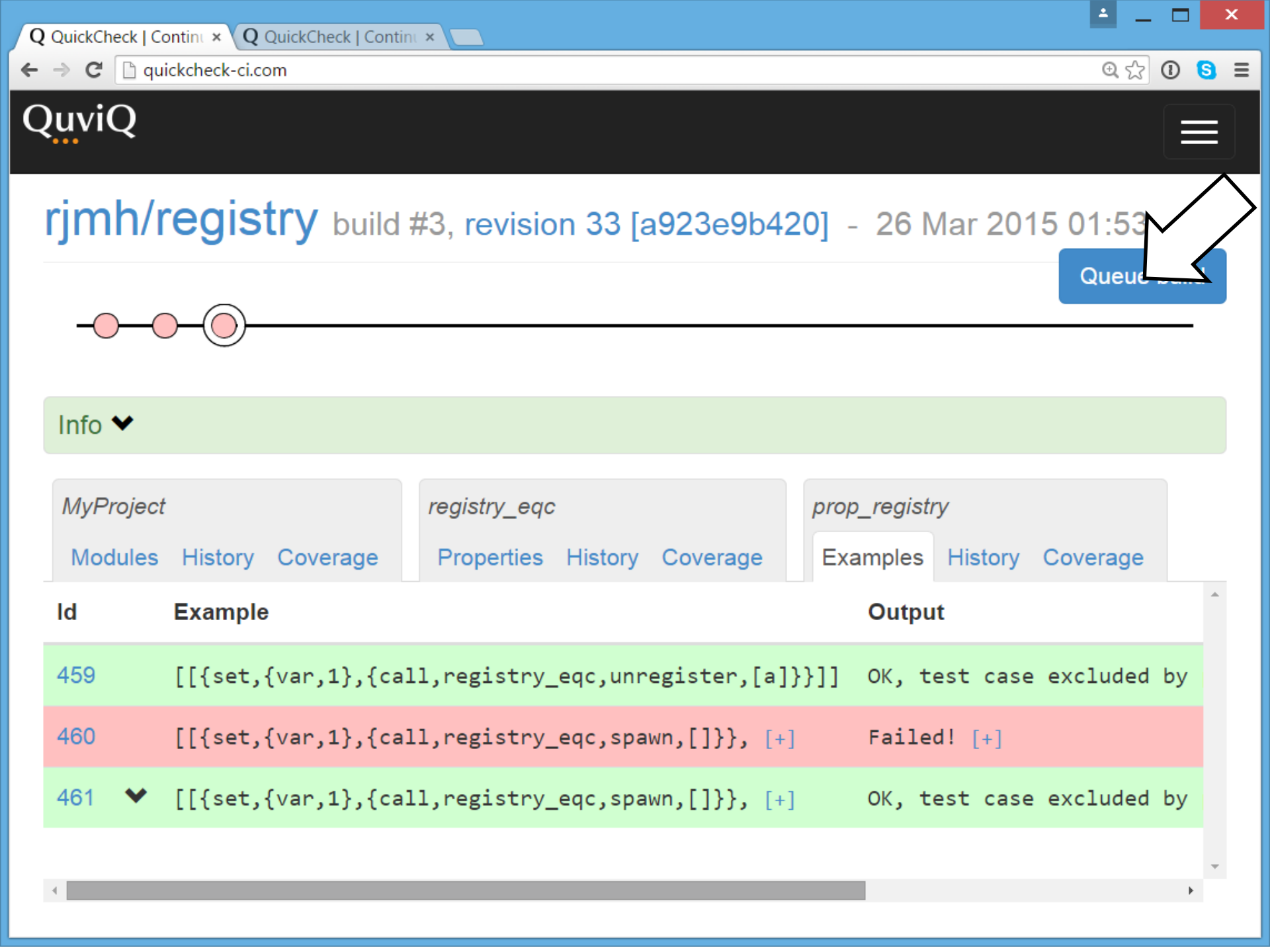
The specification of unregister

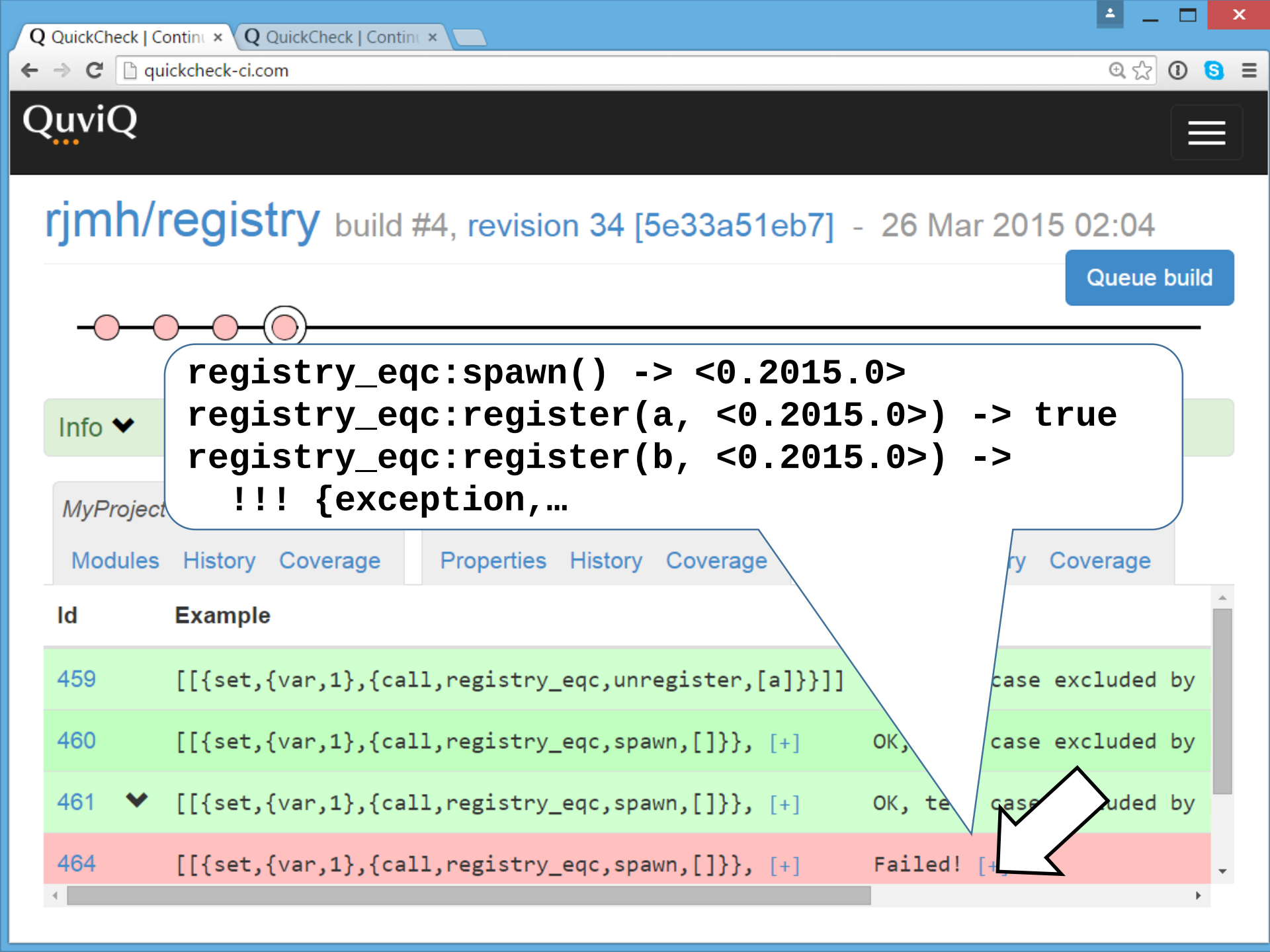
```
%% unregister

unregister_args(_) ->
  [name()].

unregister(Name) ->
  erlang:unregister(Name).

unregister_pre(S,[Name]) ->
  lists:keymember(Name,1,S#state.regs).

unregister_next(S,_,[Name]) ->
  S#state{regs=
    lists:keydelete(Name,1,S#state.regs)}.
```


QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/registry build #3, revision 33 [a923e9b420] - 26 Mar 2015 01:53
Queue build
registry_eqc:spawn() -> <0.2983.0>
registry_eqc:register(a, <0.2983.0>) -> true
registry_eqc:register(a, <0.2983.0>) ->
!!! {exception,…
Modules History Coverage
Properties History
History Coverage
Id
Example
459 [[{set,{var,1},{call,registry_eqc,unregister,[a]}}]]
460 [[{set,{var,1},{call,registry_eqc,spawn,[]}}, [+]
Failed! [+]
461 [[{set,{var,1},{call,registry_eqc,spawn,[]}}, [+]
OK, test case excluded by

Specification of register

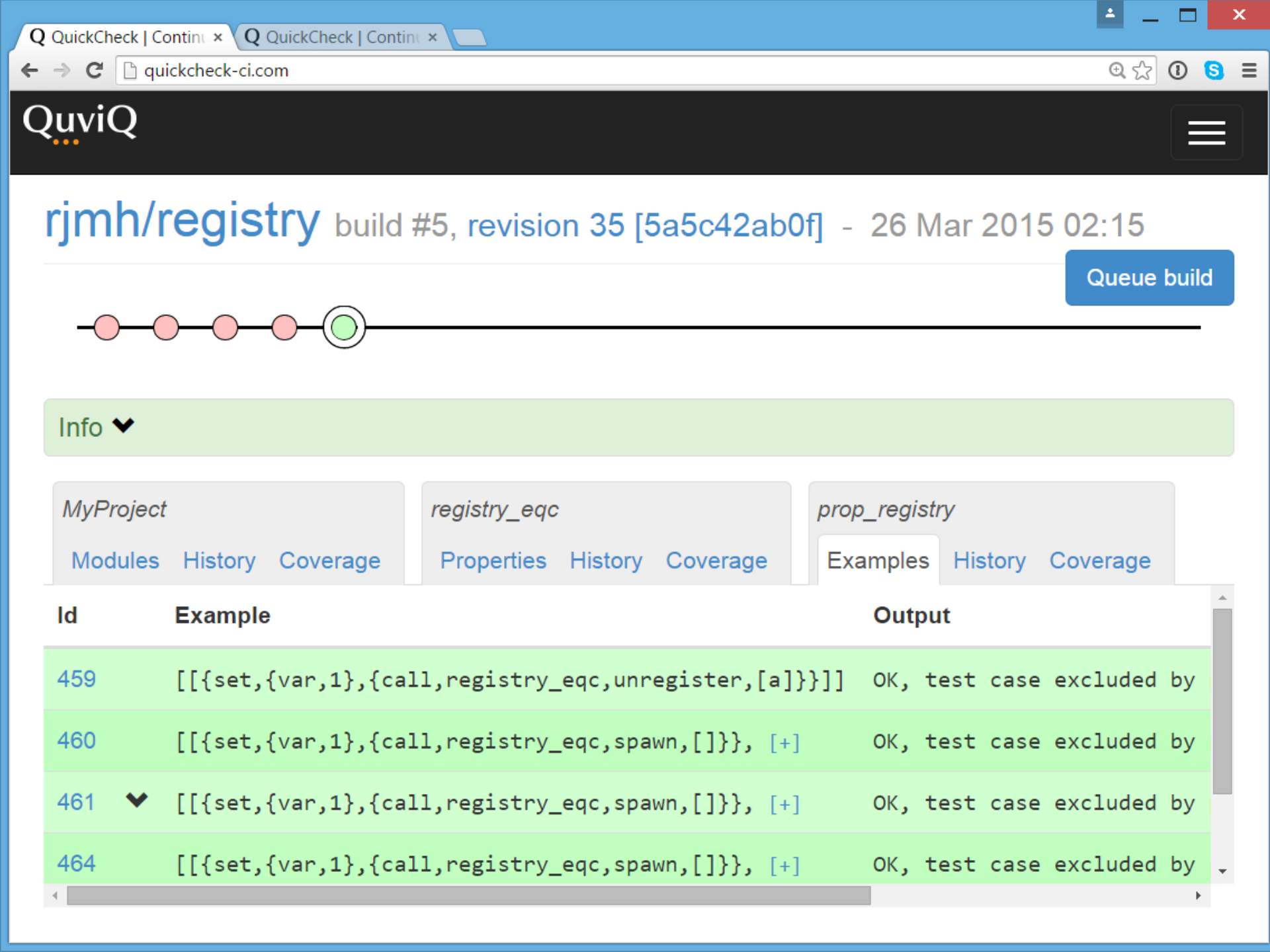
```
register_pre(S) ->
  S#state.pids /= [].

register_args(S) ->
  [name(),elements(S#state.pids)].

register(Name,Pid) ->
  erlang:register(Name,Pid).

register_next(S,_,[Name,Pid]) ->
  S#state{regs=S#state.regs++[{Name,Pid}]}.

register_pre(S,[Name,Pid]) ->
  not lists:keymember(Name,1,S#state.regs).
```


QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/registry build #3, revision 33 [a923e9b420] - 26 Mar 2015 01:53
Queue build
Info
MyProject
Modules History Coverage
registry_eqc
Properties History Coverage
prop_registry
Examples History Coverage
Id
Example
Output
459 [[{set,{var,1},{call,registry_eqc,unregister,[a]}}]] OK, test case excluded by
460 [[{set,{var,1},{call,registry_eqc,spawn,[]}}, [+] Failed! [+]
461 [[{set,{var,1},{call,registry_eqc,spawn,[]}}, [+] OK, test case excluded by


QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/registry build #4, revision 34 [5e33a51eb7] - 26 Mar 2015 02:04
Queue build
registry_eqc:spawn() -> <0.2015.0>
registry_eqc:register(a, <0.2015.0>) -> true
registry_eqc:register(b, <0.2015.0>) ->
!!! {exception,…
Info
MyProject
Modules History Coverage
Properties History Coverage
Coverage
Id
Example
459 [[{set,{var,1},{call,registry_eqc,unregister,[a]}}]]
460 [[{set,{var,1},{call,registry_eqc,spawn,[]}}, [+]
461 [[{set,{var,1},{call,registry_eqc,spawn,[]}}, [+]
464 [[{set,{var,1},{call,registry_eqc,spawn,[]}}, [+]
Failed!

Specification of register

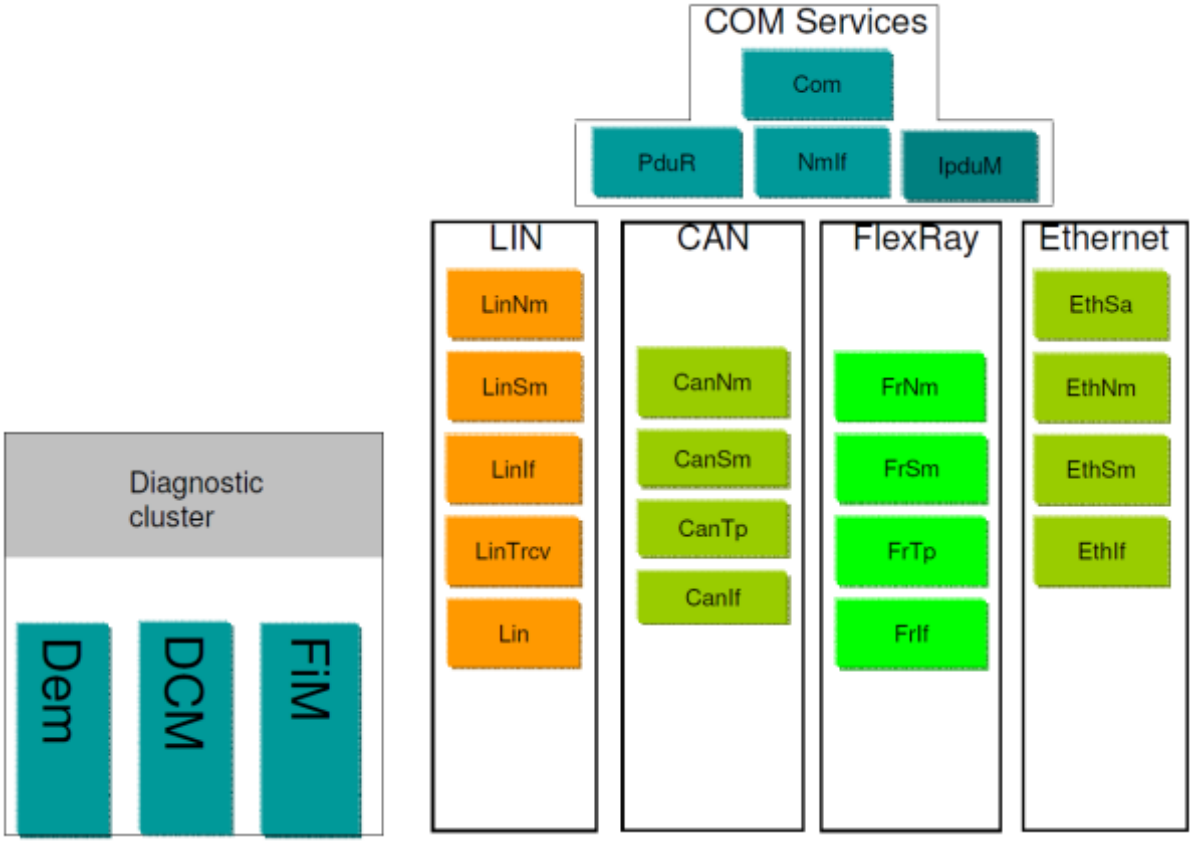
```
register_pre(S) ->
  S#state.pids /= [].

register_args(S) ->
  [name(),elements(S#state.pids)].

register(Name,Pid) ->
  erlang:register(Name,Pid).

register_next(S,_,[Name,Pid]) ->
  S#state{regs=S#state.regs++[{Name,Pid}]}.

register_pre(S,[Name,Pid]) ->
  not lists:keymember(Name,1,S#state.regs)
  andalso
  not lists:keymember(Pid,2,S#state.regs).
```


QuickCheck | Contin
QuickCheck | Contin
quickcheck-ci.com
QuviQ
rjmh/registry build #5, revision 35 [5a5c42ab0f] - 26 Mar 2015 02:15
Queue build
Info
MyProject
Modules History Coverage
registry_eqc
Properties History Coverage
prop_registry
Examples History Coverage
Id
Example
Output
459 [[{set,{var,1},{call,registry_eqc,unregister,[a]}}]] OK, test case excluded by
460 [[{set,{var,1},{call,registry_eqc,spawn,[]}}, [+] OK, test case excluded by
461 [[{set,{var,1},{call,registry_eqc,spawn,[]}}, [+] OK, test case excluded by
464 [[{set,{var,1},{call,registry_eqc,spawn,[]}}, [+] OK, test case excluded by

State machine models

- Conveniently specify the intended behaviour of stateful systems
- QuickCheck CI reports a variety of interesting test cases, and groups them sensibly
- Testing in practice involves
  - Reverse engineering of specifications (yes, really!)
  - Finding and correcting bugs in the code

Doing it for real…



3,000 pages of specifications

20,000 lines of QuickCheck

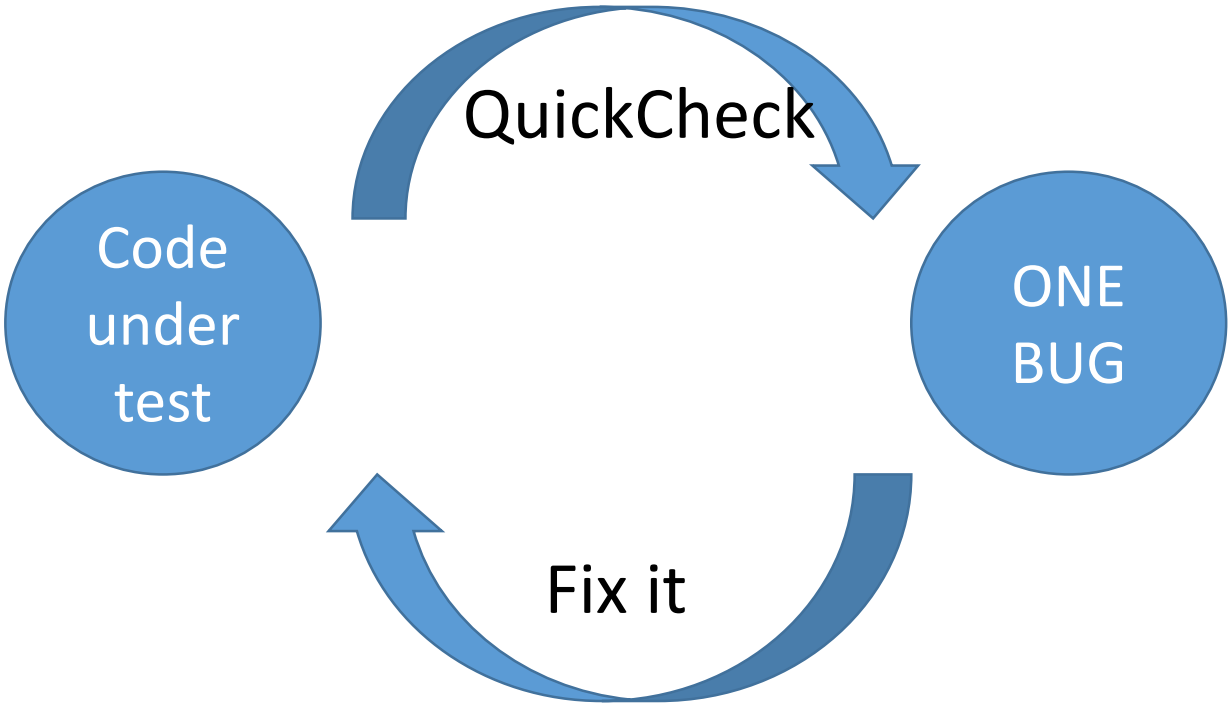
1,000,000 LOC, 6 suppliers

200 problems

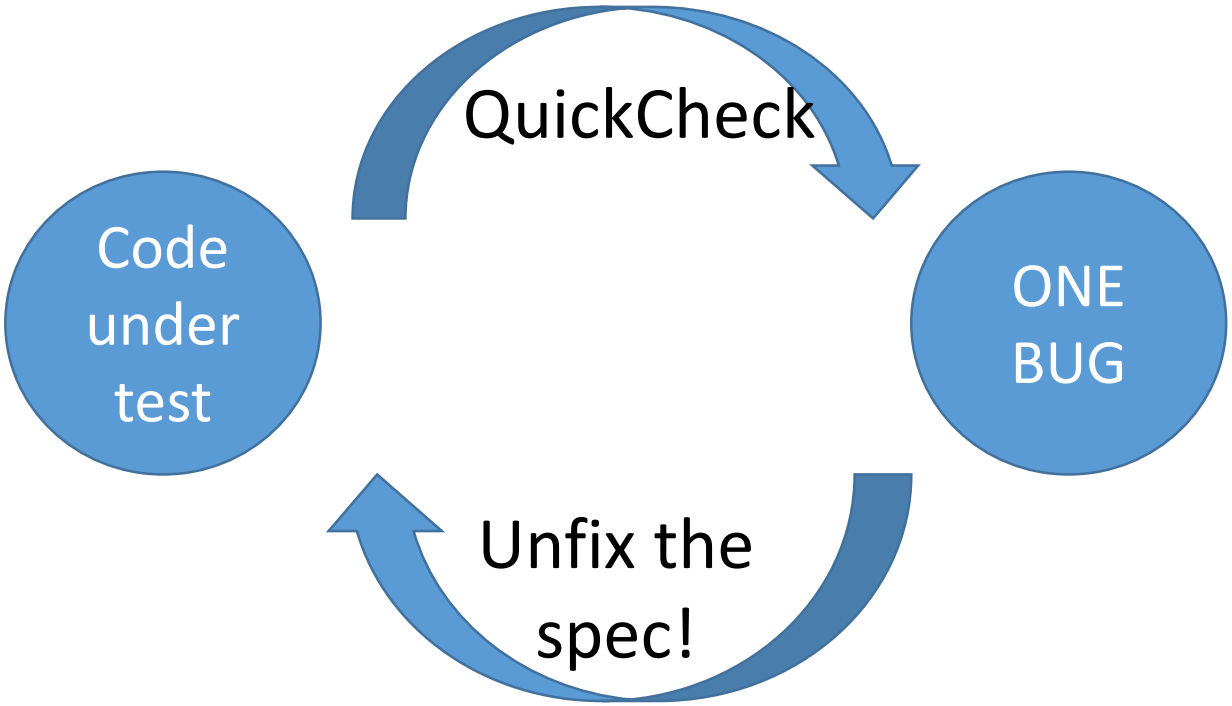
100 problems in the standard

10x shorter test code

Bugs are plural!



In practice…



”More bugs”

- Compute ”bug signatures” from failures already found
- *Focus testing effort* on areas not yet known to be buggy
- Fully automated… on the way into QuickCheck CI

In practice…

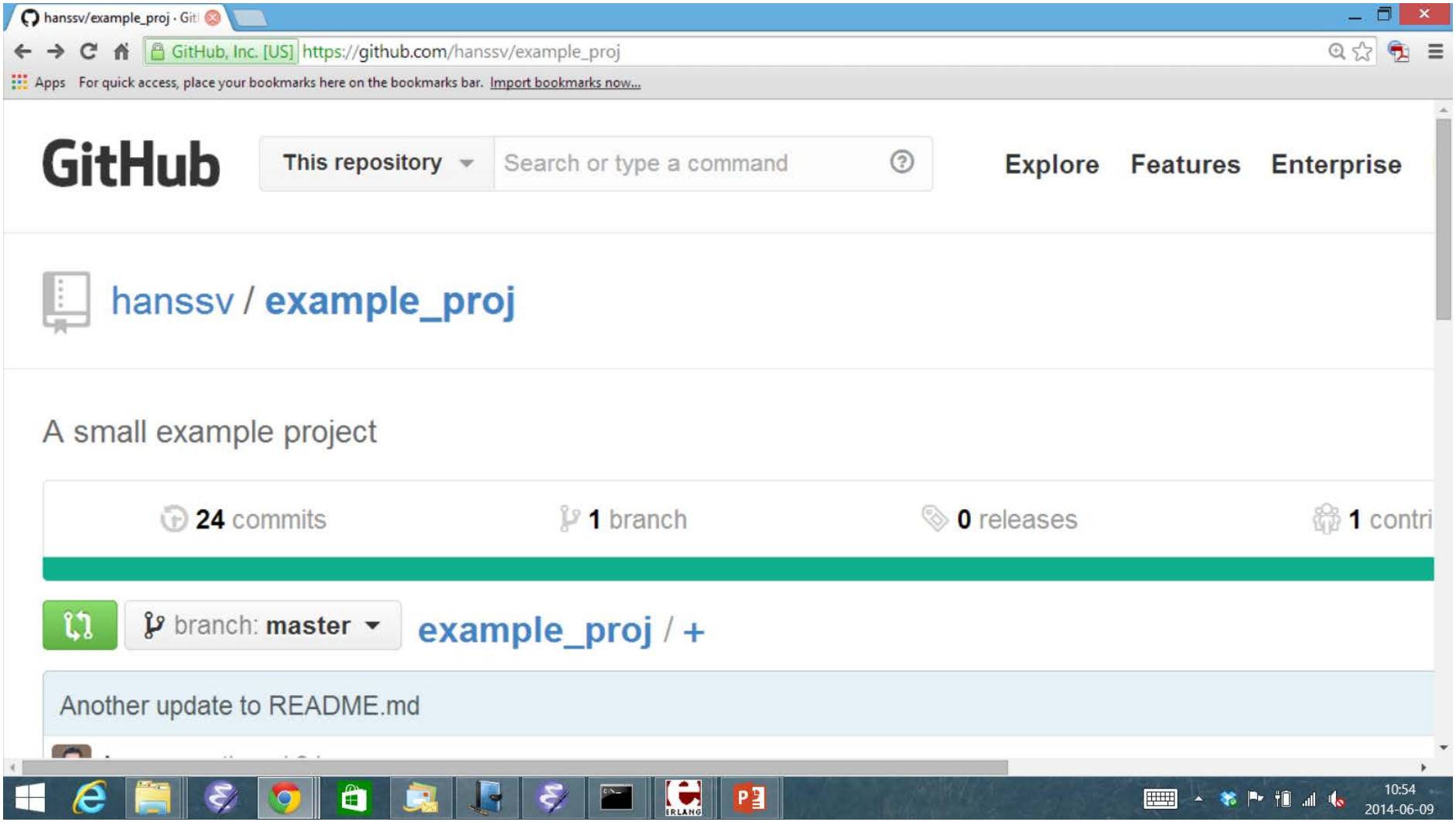
ONE BUG → **20+ BUGS**

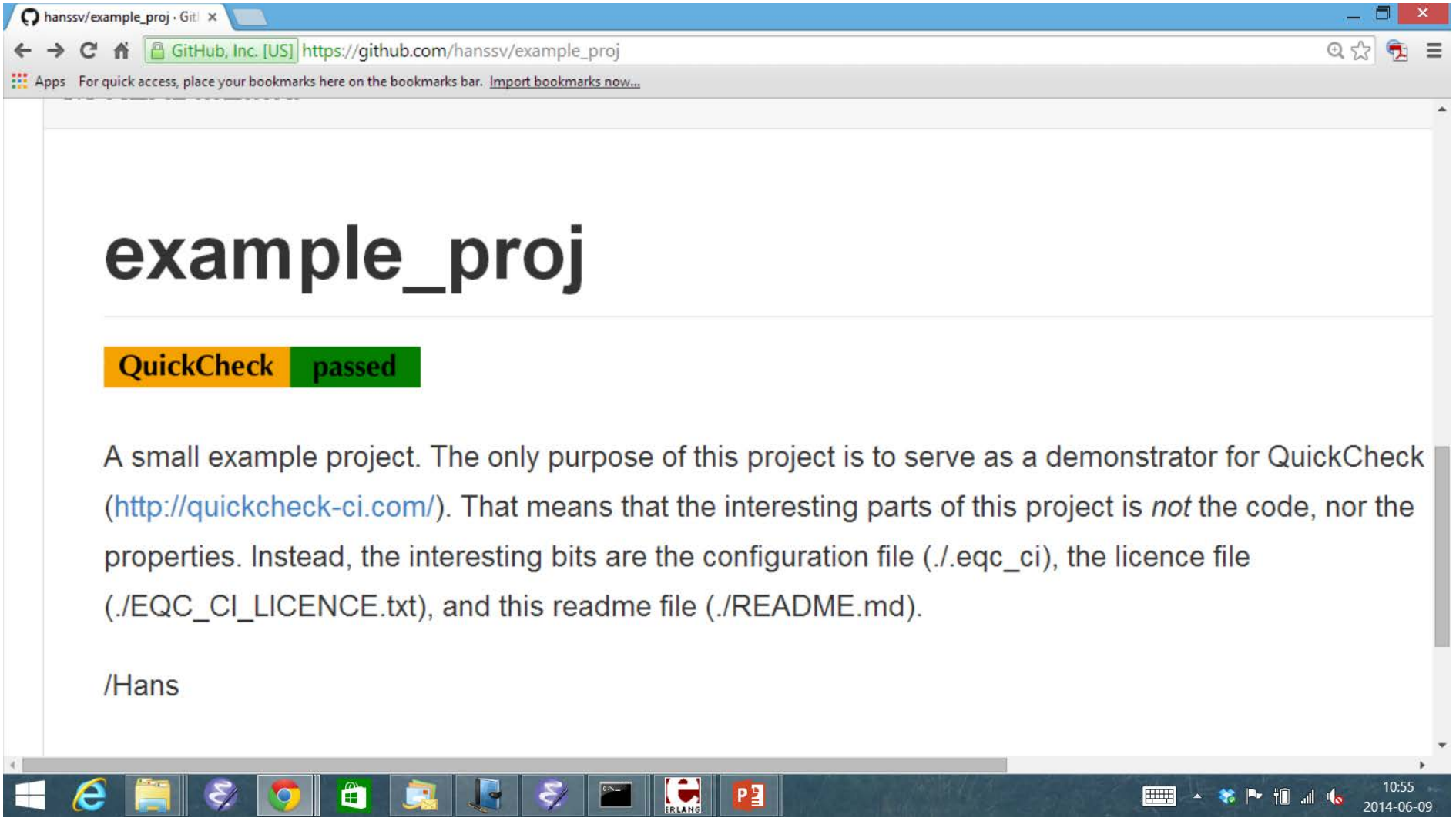
A WEEK → **A FEW HOURS**

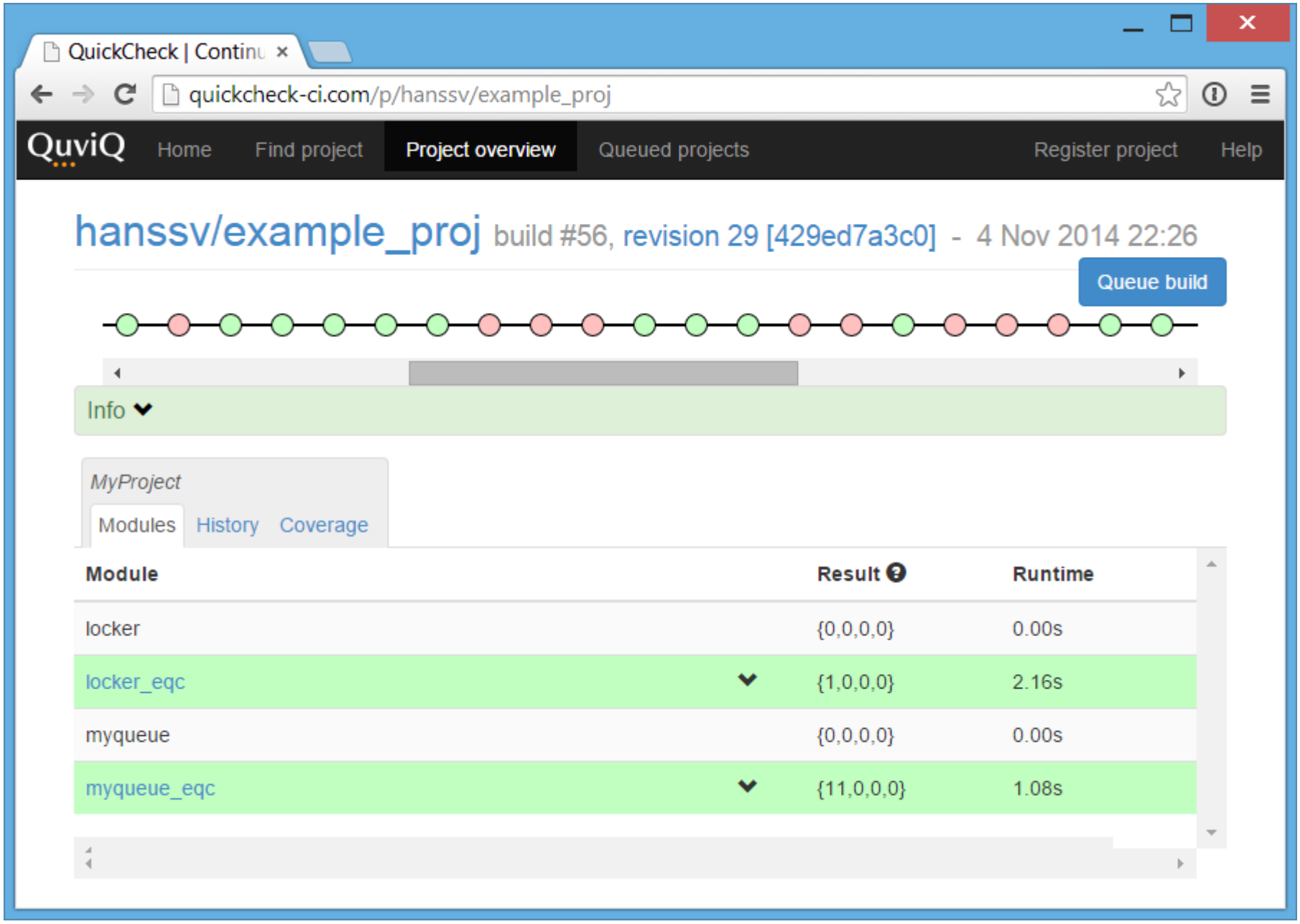
Want to try it out?

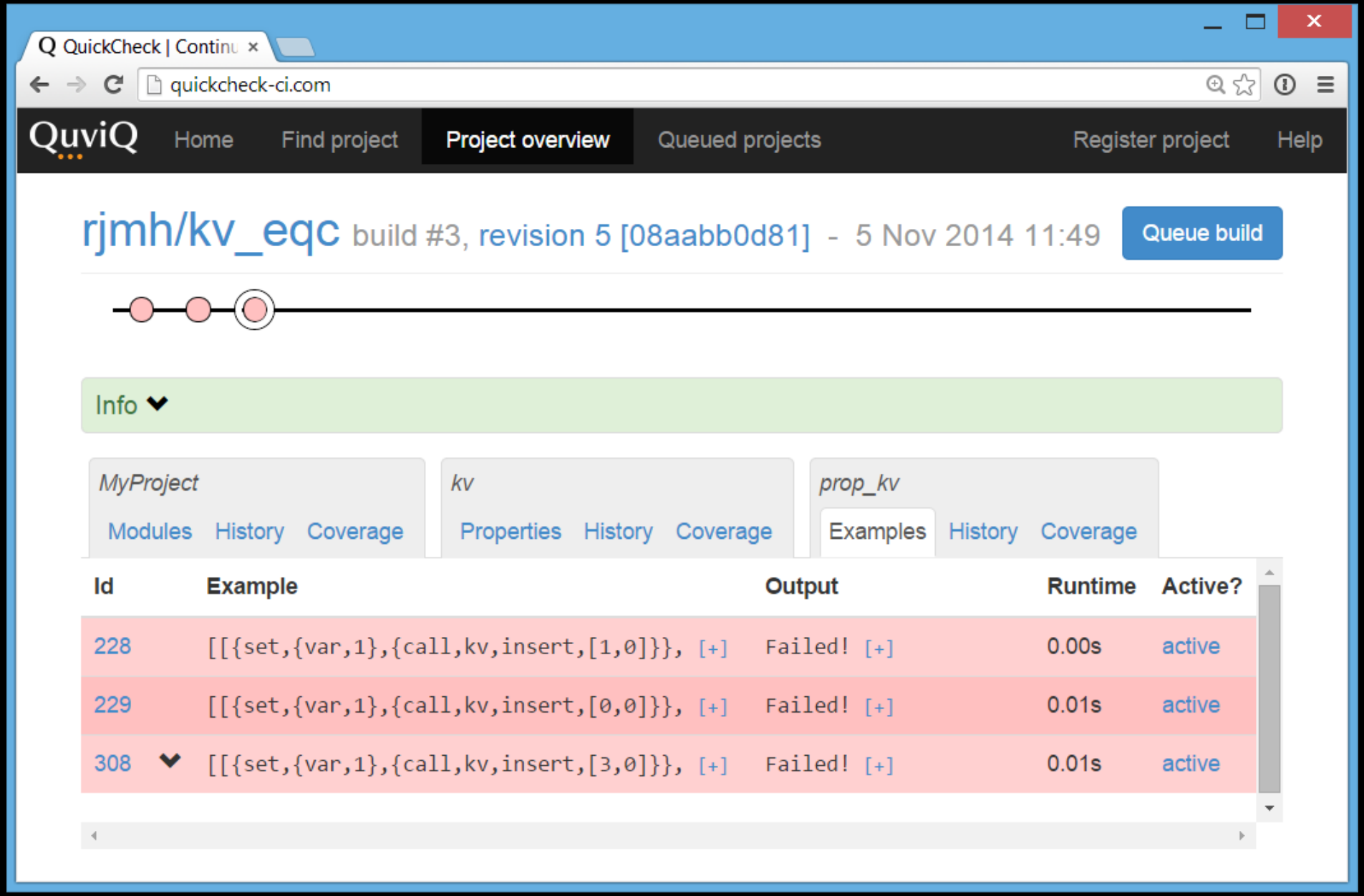
- Go to https://github.com/hanssv/example_proj
- Or clone https://github.com/rjmh/kv_eqc
  - For some QuickCheck-driven debugging fun

quickcheck-ci.com


hanssv/example_proj · Git
GitHub, Inc. [US] https://github.com/hanssv/example_proj
Apps For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
GitHub
This repository
Search or type a command
Explore
Features
Enterprise
hanssv / example_proj
A small example project
24 commits
1 branch
0 releases
1 contri
branch: master
example_proj / +
Another update to README.md
10:54
2014-06-09


hanssv/example_proj · Gitl
GitHub, Inc. [US] https://github.com/hanssv/example_proj
Apps For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
example_proj
QuickCheck passed
A small example project. The only purpose of this project is to serve as a demonstrator for QuickCheck (http://quickcheck-ci.com/). That means that the interesting parts of this project is not the code, nor the properties. Instead, the interesting bits are the configuration file (./.eqc_ci), the licence file (./EQC_CI_LICENCE.txt), and this readme file (./README.md).
/Hans
10:55
2014-06-09


QuickCheck | Continu
quickcheck-ci.com/p/hanssv/example_proj
QuviQ
Home
Find project
Project overview
Queued projects
Register project
Help
hanssv/example_proj build #56, revision 29 [429ed7a3c0] - 4 Nov 2014 22:26
Queue build
Info
MyProject
Modules
History
Coverage
Module
Result
Runtime
locker
{0,0,0,0}
0.00s
locker_eqc
{1,0,0,0}
2.16s
myqueue
{0,0,0,0}
0.00s
myqueue_eqc
{11,0,0,0}
1.08s


QuickCheck | Continu
quickcheck-ci.com
QuviQ
Home
Find project
Project overview
Queued projects
Register project
Help
rjmh/kv_eqc build #3, revision 5 [08aabb0d81] - 5 Nov 2014 11:49
Queue build
Info
MyProject
Modules History Coverage
kv
Properties History Coverage
prop_kv
Examples History Coverage
Id
Example
Output
Runtime
Active?
228 [[{set,{var,1},{call,kv,insert,[1,0]}}, [+] Failed! [+] 0.00s active
229 [[{set,{var,1},{call,kv,insert,[0,0]}}, [+] Failed! [+] 0.01s active
308 [[{set,{var,1},{call,kv,insert,[3,0]}}, [+] Failed! [+] 0.01s active