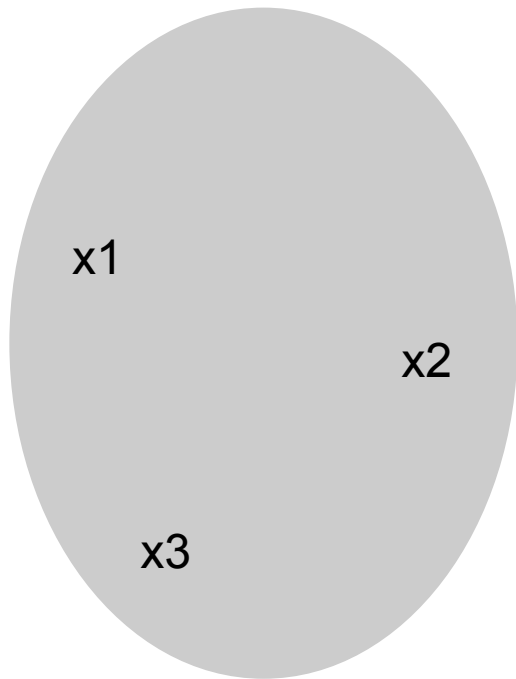


# Property-Based Testing

(QuickCheck)

# Unit Tests



"Program testing can be used to show the presence of bugs, but never to show their absence!"

## Unit Tests

- Usually, only a tiny part of the domain is tested
- Are you sure x1, x2, and x3 the right choices ?

# What is Property-Based Testing ?

**tl;dr:** Put the machine to work !

A way to:

- specify a domain to pick from *at random*
- express *properties* of programs
- run *arbitrary* numbers of tests

# Example

1. Define how to generate test data

`X : integer, L : list(integer)`

`# instead of: X = 2, L = [1,2,3]`

2. Define a property that should hold

`prop : not_member(X, delete(X, L))`

`# instead of: 2 in delete(2, [1,2,3])`

3. Run as many tests as you wish

`check(prop, 100000)`

# Shrinking

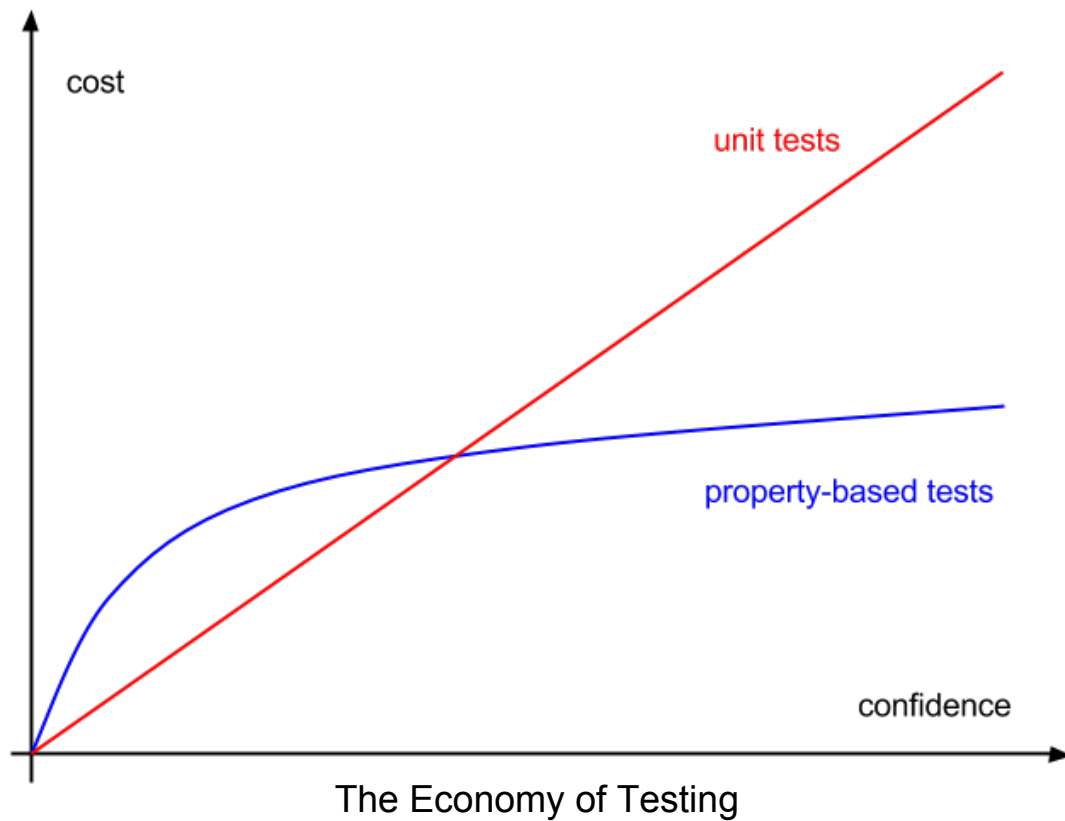
When a test case fails, QuickCheck will search for a simpler but similar test case that also fails

Useful for debugging !

# Demo

```
1 %%% How to run
2 %%% Unit tests: listdel:test().
3 %%% properties: proper:quickcheck(listdel:prop_delete()).
4
5 -module(listdel).
6
7 -export([delete_all/2]).
8 -include_lib("proper/include/proper.hrl").
9 -include_lib("eunit/include/eunit.hrl").
10
11 -import(lists,[seq/2]).
12
13
14 %%% A lists delete implementation
15 -spec delete_all(?, list()) -> list().
16 delete_all(X, L) ->
17     delete_all(X, L, []).
18
19
20 delete_all(_, [], Acc) ->
21     lists:reverse(Acc);
22 delete_all(X, [_:Rest], Acc) ->
23     %%lists:reverse(Acc) ++ Rest;
24     delete_all(X, Rest, Acc);
25 delete_all(X, [_:Rest], Acc) ->
26     delete_all(X, Rest, [X|Acc]).
27
28
29 %%% Unit tests
30 eunit:test(listdel).
31 delete_test() ->
32     [?assertEqual(delete_all([], []), []),
33      ?assertEqual(delete_all([1,2,3,4], [2,3,4]),
34                   [1,2,3,4]),
35      ?assertEqual(delete_all([5,[1,2,3,4]], [1,2,3,4]),
36                   [5,-4,-3,-2,-1]),
37      ?assertEqual(delete_all([100, seq(1,100)], seq(1,99))
38                   [],
39
40
41 %%% proper:quickcheck(listdel:prop_delete(1, 10)).
42 %%% For any integer x and any list of integers l,
43 %%% if I delete x from l, then l does not contain x
44 prop_delete() ->
45     ?FORALL(I, {1..100},
46             {integer(), list(integer())}, % variables
47             not lists:member(X, delete_all(X, L))). % property
48
49
50 %%% some with stars
51 prop_delete2() ->
52     ?FORALL(I, {1..100},
53             {integer(), list(integer())},
54             collect([I, I, not lists:member(X, delete_all(X, L))]).
55
56
57 %%% Generate more relevant test data:
58 %%% non-empty lists and only delete actual list elements
59 prop_delete3() ->
60     ?FORALL(C,
61             list(integer()),
62             listdel:delete_all(C, C)).
63
64
65 %%% List of all test cases
66 listdel_test_cases = [
67     {1, 1, 1},
68     {1, 2, 1},
69     {1, 3, 1},
70     {1, 4, 1},
71     {1, 5, 1},
72     {1, 6, 1},
73     {1, 7, 1},
74     {1, 8, 1},
75     {1, 9, 1},
76     {1, 10, 1},
77     {1, 11, 1},
78     {1, 12, 1},
79     {1, 13, 1},
80     {1, 14, 1},
81     {1, 15, 1},
82     {1, 16, 1},
83     {1, 17, 1},
84     {1, 18, 1},
85     {1, 19, 1},
86     {1, 20, 1},
87     {1, 21, 1},
88     {1, 22, 1},
89     {1, 23, 1},
90     {1, 24, 1},
91     {1, 25, 1},
92     {1, 26, 1},
93     {1, 27, 1},
94     {1, 28, 1},
95     {1, 29, 1},
96     {1, 30, 1},
97     {1, 31, 1},
98     {1, 32, 1},
99     {1, 33, 1},
100    {1, 34, 1},
101    {1, 35, 1},
102    {1, 36, 1},
103    {1, 37, 1},
104    {1, 38, 1},
105    {1, 39, 1},
106    {1, 40, 1},
107    {1, 41, 1},
108    {1, 42, 1},
109    {1, 43, 1},
110    {1, 44, 1},
111    {1, 45, 1},
112    {1, 46, 1},
113    {1, 47, 1},
114    {1, 48, 1},
115    {1, 49, 1},
116    {1, 50, 1},
117    {1, 51, 1},
118    {1, 52, 1},
119    {1, 53, 1},
120    {1, 54, 1},
121    {1, 55, 1},
122    {1, 56, 1},
123    {1, 57, 1},
124    {1, 58, 1},
125    {1, 59, 1},
126    {1, 60, 1},
127    {1, 61, 1},
128    {1, 62, 1},
129    {1, 63, 1},
130    {1, 64, 1},
131    {1, 65, 1},
132    {1, 66, 1},
133    {1, 67, 1},
134    {1, 68, 1},
135    {1, 69, 1},
136    {1, 70, 1},
137    {1, 71, 1},
138    {1, 72, 1},
139    {1, 73, 1},
140    {1, 74, 1},
141    {1, 75, 1},
142    {1, 76, 1},
143    {1, 77, 1},
144    {1, 78, 1},
145    {1, 79, 1},
146    {1, 80, 1},
147    {1, 81, 1},
148    {1, 82, 1},
149    {1, 83, 1},
150    {1, 84, 1},
151    {1, 85, 1},
152    {1, 86, 1},
153    {1, 87, 1},
154    {1, 88, 1},
155    {1, 89, 1},
156    {1, 90, 1},
157    {1, 91, 1},
158    {1, 92, 1},
159    {1, 93, 1},
160    {1, 94, 1},
161    {1, 95, 1},
162    {1, 96, 1},
163    {1, 97, 1},
164    {1, 98, 1},
165    {1, 99, 1},
166    {1, 100, 1},
167    {1, 101, 1},
168    {1, 102, 1},
169    {1, 103, 1},
170    {1, 104, 1},
171    {1, 105, 1},
172    {1, 106, 1},
173    {1, 107, 1},
174    {1, 108, 1},
175    {1, 109, 1},
176    {1, 110, 1},
177    {1, 111, 1},
178    {1, 112, 1},
179    {1, 113, 1},
180    {1, 114, 1},
181    {1, 115, 1},
182    {1, 116, 1},
183    {1, 117, 1},
184    {1, 118, 1},
185    {1, 119, 1},
186    {1, 120, 1},
187    {1, 121, 1},
188    {1, 122, 1},
189    {1, 123, 1},
190    {1, 124, 1},
191    {1, 125, 1},
192    {1, 126, 1},
193    {1, 127, 1},
194    {1, 128, 1},
195    {1, 129, 1},
196    {1, 130, 1},
197    {1, 131, 1},
198    {1, 132, 1},
199    {1, 133, 1},
200    {1, 134, 1},
201    {1, 135, 1},
202    {1, 136, 1},
203    {1, 137, 1},
204    {1, 138, 1},
205    {1, 139, 1},
206    {1, 140, 1},
207    {1, 141, 1},
208    {1, 142, 1},
209    {1, 143, 1},
210    {1, 144, 1},
211    {1, 145, 1},
212    {1, 146, 1},
213    {1, 147, 1},
214    {1, 148, 1},
215    {1, 149, 1},
216    {1, 150, 1},
217    {1, 151, 1},
218    {1, 152, 1},
219    {1, 153, 1},
220    {1, 154, 1},
221    {1, 155, 1},
222    {1, 156, 1},
223    {1, 157, 1},
224    {1, 158, 1},
225    {1, 159, 1},
226    {1, 160, 1},
227    {1, 161, 1},
228    {1, 162, 1},
229    {1, 163, 1},
230    {1, 164, 1},
231    {1, 165, 1},
232    {1, 166, 1},
233    {1, 167, 1},
234    {1, 168, 1},
235    {1, 169, 1},
236    {1, 170, 1},
237    {1, 171, 1},
238    {1, 172, 1},
239    {1, 173, 1},
240    {1, 174, 1},
241    {1, 175, 1},
242    {1, 176, 1},
243    {1, 177, 1},
244    {1, 178, 1},
245    {1, 179, 1},
246    {1, 180, 1},
247    {1, 181, 1},
248    {1, 182, 1},
249    {1, 183, 1},
250    {1, 184, 1},
251    {1, 185, 1},
252    {1, 186, 1},
253    {1, 187, 1},
254    {1, 188, 1},
255    {1, 189, 1},
256    {1, 190, 1},
257    {1, 191, 1},
258    {1, 192, 1},
259    {1, 193, 1},
260    {1, 194, 1},
261    {1, 195, 1},
262    {1, 196, 1},
263    {1, 197, 1},
264    {1, 198, 1},
265    {1, 199, 1},
266    {1, 200, 1},
267    {1, 201, 1},
268    {1, 202, 1},
269    {1, 203, 1},
270    {1, 204, 1},
271    {1, 205, 1},
272    {1, 206, 1},
273    {1, 207, 1},
274    {1, 208, 1},
275    {1, 209, 1},
276    {1, 210, 1},
277    {1, 211, 1},
278    {1, 212, 1},
279    {1, 213, 1},
280    {1, 214, 1},
281    {1, 215, 1},
282    {1, 216, 1},
283    {1, 217, 1},
284    {1, 218, 1},
285    {1, 219, 1},
286    {1, 220, 1},
287    {1, 221, 1},
288    {1, 222, 1},
289    {1, 223, 1},
290    {1, 224, 1},
291    {1, 225, 1},
292    {1, 226, 1},
293    {1, 227, 1},
294    {1, 228, 1},
295    {1, 229, 1},
296    {1, 230, 1},
297    {1, 231, 1},
298    {1, 232, 1},
299    {1, 233, 1},
300    {1, 234, 1},
301    {1, 235, 1},
302    {1, 236, 1},
303    {1, 237, 1},
304    {1, 238, 1},
305    {1, 239, 1},
306    {1, 240, 1},
307    {1, 241, 1},
308    {1, 242, 1},
309    {1, 243, 1},
310    {1, 244, 1},
311    {1, 245, 1},
312    {1, 246, 1},
313    {1, 247, 1},
314    {1, 248, 1},
315    {1, 249, 1},
316    {1, 250, 1},
317    {1, 251, 1},
318    {1, 252, 1},
319    {1, 253, 1},
320    {1, 254, 1},
321    {1, 255, 1},
322    {1, 256, 1},
323    {1, 257, 1},
324    {1, 258, 1},
325    {1, 259, 1},
326    {1, 260, 1},
327    {1, 261, 1},
328    {1, 262, 1},
329    {1, 263, 1},
330    {1, 264, 1},
331    {1, 265, 1},
332    {1, 266, 1},
333    {1, 267, 1},
334    {1, 268, 1},
335    {1, 269, 1},
336    {1, 270, 1},
337    {1, 271, 1},
338    {1, 272, 1},
339    {1, 273, 1},
340    {1, 274, 1},
341    {1, 275, 1},
342    {1, 276, 1},
343    {1, 277, 1},
344    {1, 278, 1},
345    {1, 279, 1},
346    {1, 280, 1},
347    {1, 281, 1},
348    {1, 282, 1},
349    {1, 283, 1},
350    {1, 284, 1},
351    {1, 285, 1},
352    {1, 286, 1},
353    {1, 287, 1},
354    {1, 288, 1},
355    {1, 289, 1},
356    {1, 290, 1},
357    {1, 291, 1},
358    {1, 292, 1},
359    {1, 293, 1},
360    {1, 294, 1},
361    {1, 295, 1},
362    {1, 296, 1},
363    {1, 297, 1},
364    {1, 298, 1},
365    {1, 299, 1},
366    {1, 300, 1},
367    {1, 301, 1},
368    {1, 302, 1},
369    {1, 303, 1},
370    {1, 304, 1},
371    {1, 305, 1},
372    {1, 306, 1},
373    {1, 307, 1},
374    {1, 308, 1},
375    {1, 309, 1},
376    {1, 310, 1},
377    {1, 311, 1},
378    {1, 312, 1},
379    {1, 313, 1},
380    {1, 314, 1},
381    {1, 315, 1},
382    {1, 316, 1},
383    {1, 317, 1},
384    {1, 318, 1},
385    {1, 319, 1},
386    {1, 320, 1},
387    {1, 321, 1},
388    {1, 322, 1},
389    {1, 323, 1},
390    {1, 324, 1},
391    {1, 325, 1},
392    {1, 326, 1},
393    {1, 327, 1},
394    {1, 328, 1},
395    {1, 329, 1},
396    {1, 330, 1},
397    {1, 331, 1},
398    {1, 332, 1},
399    {1, 333, 1},
400    {1, 334, 1},
401    {1, 335, 1},
402    {1, 336, 1},
403    {1, 337, 1},
404    {1, 338, 1},
405    {1, 339, 1},
406    {1, 340, 1},
407    {1, 341, 1},
408    {1, 342, 1},
409    {1, 343, 1},
410    {1, 344, 1},
411    {1, 345, 1},
412    {1, 346, 1},
413    {1, 347, 1},
414    {1, 348, 1},
415    {1, 349, 1},
416    {1, 350, 1},
417    {1, 351, 1},
418    {1, 352, 1},
419    {1, 353, 1},
420    {1, 354, 1},
421    {1, 355, 1},
422    {1, 356, 1},
423    {1, 357, 1},
424    {1, 358, 1},
425    {1, 359, 1},
426    {1, 360, 1},
427    {1, 361, 1},
428    {1, 362, 1},
429    {1, 363, 1},
430    {1, 364, 1},
431    {1, 365, 1},
432    {1, 366, 1},
433    {1, 367, 1},
434    {1, 368, 1},
435    {1, 369, 1},
436    {1, 370, 1},
437    {1, 371, 1},
438    {1, 372, 1},
439    {1, 373, 1},
440    {1, 374, 1},
441    {1, 375, 1},
442    {1, 376, 1},
443    {1, 377, 1},
444    {1, 378, 1},
445    {1, 379, 1},
446    {1, 380, 1},
447    {1, 381, 1},
448    {1, 382, 1},
449    {1, 383, 1},
450    {1, 384, 1},
451    {1, 385, 1},
452    {1, 386, 1},
453    {1, 387, 1},
454    {1, 388, 1},
455    {1, 389, 1},
456    {1, 390, 1},
457    {1, 391, 1},
458    {1, 392, 1},
459    {1, 393, 1},
460    {1, 394, 1},
461    {1, 395, 1},
462    {1, 396, 1},
463    {1, 397, 1},
464    {1, 398, 1},
465    {1, 399, 1},
466    {1, 400, 1},
467    {1, 401, 1},
468    {1, 402, 1},
469    {1, 403, 1},
470    {1, 404, 1},
471    {1, 405, 1},
472    {1, 406, 1},
473    {1, 407, 1},
474    {1, 408, 1},
475    {1, 409, 1},
476    {1, 410, 1},
477    {1, 411, 1},
478    {1, 412, 1},
479    {1, 413, 1},
480    {1, 414, 1},
481    {1, 415, 1},
482    {1, 416, 1},
483    {1, 417, 1},
484    {1, 418, 1},
485    {1, 419, 1},
486    {1, 420, 1},
487    {1, 421, 1},
488    {1, 422, 1},
489    {1, 423, 1},
490    {1, 424, 1},
491    {1, 425, 1},
492    {1, 426, 1},
493    {1, 427, 1},
494    {1, 428, 1},
495    {1, 429, 1},
496    {1, 430, 1},
497    {1, 431, 1},
498    {1, 432, 1},
499    {1, 433, 1},
500    {1, 434, 1},
501    {1, 435, 1},
502    {1, 436, 1},
503    {1, 437, 1},
504    {1, 438, 1},
505    {1, 439, 1},
506    {1, 440, 1},
507    {1, 441, 1},
508    {1, 442, 1},
509    {1, 443, 1},
510    {1, 444, 1},
511    {1, 445, 1},
512    {1, 446, 1},
513    {1, 447, 1},
514    {1, 448, 1},
515    {1, 449, 1},
516    {1, 450, 1},
517    {1, 451, 1},
518    {1, 452, 1},
519    {1, 453, 1},
520    {1, 454, 1},
521    {1, 455, 1},
522    {1, 456, 1},
523    {1, 457, 1},
524    {1, 458, 1},
525    {1, 459, 1},
526    {1, 460, 1},
527    {1, 461, 1},
528    {1, 462, 1},
529    {1, 463, 1},
530    {1, 464, 1},
531    {1, 465, 1},
532    {1, 466, 1},
533    {1, 467, 1},
534    {1, 468, 1},
535    {1, 469, 1},
536    {1, 470, 1},
537    {1, 471, 1},
538    {1, 472, 1},
539    {1, 473, 1},
540    {1, 474, 1},
541    {1, 475, 1},
542    {1, 476, 1},
543    {1, 477, 1},
544    {1, 478, 1},
545    {1, 479, 1},
546    {1, 480, 1},
547    {1, 481, 1},
548    {1, 482, 1},
549    {1, 483, 1},
550    {1, 484, 1},
551    {1, 485, 1},
552    {1, 486, 1},
553    {1, 487, 1},
554    {1, 488, 1},
555    {1, 489, 1},
556    {1, 490, 1},
557    {1, 491, 1},
558    {1, 492, 1},
559    {1, 493, 1},
560    {1, 494, 1},
561    {1, 495, 1},
562    {1, 496, 1},
563    {1, 497, 1},
564    {1, 498, 1},
565    {1, 499, 1},
566    {1, 500, 1},
567    {1, 501, 1},
568    {1, 502, 1},
569    {1, 503, 1},
570    {1, 504, 1},
571    {1, 505, 1},
572    {1, 506, 1},
573    {1, 507, 1},
574    {1, 508, 1},
575    {1, 509, 1},
576    {1, 510, 1},
577    {1, 511, 1},
578    {1, 512, 1},
579    {1, 513, 1},
580    {1, 514, 1},
581    {1, 515, 1},
582    {1, 516, 1},
583    {1, 517, 1},
584    {1, 518, 1},
585    {1, 519, 1},
586    {1, 520, 1},
587    {1, 521, 1},
588    {1, 522, 1},
589    {1, 523, 1},
590    {1, 524, 1},
591    {1, 525, 1},
592    {1, 526, 1},
593    {1, 527, 1},
594    {1, 528, 1},
595    {1, 529, 1},
596    {1, 530, 1},
597    {1, 531, 1},
598    {1, 532, 1},
599    {1, 533, 1},
600    {1, 534, 1},
601    {1, 535, 1},
602    {1, 536, 1},
603    {1, 537, 1},
604    {1, 538, 1},
605    {1, 539, 1},
606    {1, 540, 1},
607    {1, 541, 1},
608    {1, 542, 1},
609    {1, 543, 1},
610    {1, 544, 1},
611    {1, 545, 1},
612    {1, 546, 1},
613    {1, 547, 1},
614    {1, 548, 1},
615    {1, 549, 1},
616    {1, 550, 1},
617    {1, 551, 1},
618    {1, 552, 1},
619    {1, 553, 1},
620    {1, 554, 1},
621    {1, 555, 1},
622    {1, 556, 1},
623    {1, 557, 1},
624    {1, 558, 1},
625    {1, 559, 1},
626    {1, 560, 1},
627    {1, 561, 1},
628    {1, 562, 1},
629    {1, 563, 1},
630    {1, 564, 1},
631    {1, 565, 1},
632    {1, 566, 1},
633    {1, 567, 1},
634    {1, 568, 1},
635    {1, 569, 1},
636    {1, 570, 1},
637    {1, 571, 1},
638    {1, 572, 1},
639    {1, 573, 1},
640    {1, 574, 1},
641    {1, 575, 1},
642    {1, 576, 1},
643    {1, 577, 1},
644    {1, 578, 1},
645    {1, 579, 1},
646    {1, 580, 1},
647    {1, 581, 1},
648    {1, 582, 1},
649    {1, 583, 1},
650    {1, 584, 1},
651    {1, 585, 1},
652    {1, 586, 1},
653    {1, 587, 1},
654    {1, 588, 1},
655    {1, 589, 1},
656    {1, 590, 1},
657    {1, 591, 1},
658    {1, 592, 1},
659    {1, 593, 1},
660    {1, 594, 1},
661    {1, 595, 1},
662    {1, 596, 1},
663    {1, 597, 1},
664    {1, 598, 1},
665    {1, 599, 1},
666    {1, 600, 1},
667    {1, 601, 1},
668    {1, 602, 1},
669    {1, 603, 1},
670    {1, 604, 1},
671    {1, 605, 1},
672    {1, 606, 1},
673    {1, 607, 1},
674    {1, 608, 1},
675    {1, 609, 1},
676    {1, 610, 1},
677    {1, 611, 1},
678    {1, 612, 1},
679    {1, 613, 1},
680    {1, 614, 1},
681    {1, 615, 1},
682    {1, 616, 1},
683    {1, 617, 1},
684    {1, 618, 1},
685    {1, 619, 1},
686    {1, 620, 1},
687    {1, 621, 1},
688    {1, 622, 1},
689    {1, 623, 1},
690    {1, 624, 1},
691    {1, 625, 1},
692    {1, 626, 1},
693    {1, 627, 1},
694    {1, 628, 1},
695    {1, 629, 1},
696    {1, 630, 1},
697    {1, 631, 1},
698    {1, 632, 1},
699    {1, 633, 1},
700    {1, 634, 1},
701    {1, 635, 1},
702    {1, 636, 1},
703    {1, 637, 1},
704    {1, 638, 1},
705    {1, 639, 1},
706    {1, 640, 1},
707    {1, 641, 1},
708    {1, 642, 1},
709    {1, 643, 1},
710    {1, 644, 1},
711    {1, 645, 1},
712    {1, 646, 1},
713    {1, 647, 1},
714    {1, 648, 1},
715    {1, 649, 1},
716    {1, 650, 1},
717    {1, 651, 1},
718    {1, 652, 1},
719    {1, 653, 1},
720    {1, 654, 1},
721    {1, 655, 1},
722    {1, 656, 1},
723    {1, 657, 1},
724    {1, 658, 1},
725    {1, 659, 1},
726    {1, 660, 1},
727    {1, 661, 1},
728    {1, 662, 1},
729    {1, 663, 1},
730    {1, 664, 1},
731    {1, 665, 1},
732    {1, 666, 1},
733    {1, 667, 1},
734    {1, 668, 1},
735    {1, 669, 1},
736    {1, 670, 1},
737    {1, 671, 1},
738    {1, 672, 1},
739    {1, 673, 1},
740    {1, 674, 1},
741    {1, 675, 1},
742    {1, 676, 1},
743    {1, 677, 1},
744    {1, 678, 1},
745    {1, 679, 1},
746    {1, 680, 1},
747    {1, 681, 1},
748    {1, 682, 1},
749    {1, 683, 1},
750    {1, 684, 1},
751    {1, 685, 1},
752    {1, 686, 1},
753    {1, 687, 1},
754    {1, 688, 1},
755    {1, 689, 1},
756    {1, 690, 1},
757    {1, 691, 1},
758    {1, 692, 1},
759    {1, 693, 1},
760    {1, 694, 1},
761    {1, 695, 1},
762    {1, 696, 1},
763    {1, 697, 1},
764    {1, 698, 1},
765    {1, 699, 1},
766    {1, 700, 1},
767    {1, 701, 1},
768    {1, 702, 1},
769    {1, 703, 1},
770    {1, 704, 1},
771    {1, 705, 1},
772    {1, 706, 1},
773    {1, 707, 1},
774    {1, 708, 1},
775    {1, 709, 1},
776    {1, 710, 1},
777    {1, 711, 1},
778    {1, 712, 1},
779    {1, 713, 1},
780    {1, 714, 1},
781    {1, 715, 1},
782    {1, 716, 1},
783    {1, 717, 1},
784    {1, 718, 1},
785    {1, 719, 1},
786    {1, 720, 1},
787    {1, 721, 1},
788    {1, 722, 1},
789    {1, 723, 1},
790    {1, 724, 1},
791    {1, 725, 1},
792    {1, 726, 1},
793    {1, 727, 1},
794    {1, 728, 1},
795    {1, 729, 1},
796    {1, 730, 1},
797    {1, 731, 1},
798    {1, 732, 1},
799    {1, 733, 1},
800    {1, 734, 1},
801    {
```

# Why ?



# The Catch ?

- We test against a specification: sometimes it's the specification that is wrong !
- It can be difficult to come up with "good" properties and they can be tricky to express
- How relevant is the randomly generated test data ?  
(use stats to ensure a good test case distribution !)



# Commercial Users

- Ericsson
- Motorola
- Volvo
- Basho (Riak)

From QuviQ's website: <http://www.quviq.com/>

# The End...

Find a clone in your favorite language...

C, C++, Chicken Scheme, Clojure, Common Lisp, D, F#,  
Factor, Haskell, Io, Java, JavaScript, Node.js, Obj-C,  
OCaml, Perl, Python, Ruby, Scala, Scheme, Smalltalk,  
SML.

<http://en.wikipedia.org/wiki/QuickCheck>