

THE TROUBLE WITH TESTING

a brief comic meant
to start some
brainstorming, by
kirk israel

Good testing is
tough!

The problem, here, is very fundamental. In order to mechanically prove that a program corresponds to some spec, the spec itself needs to be extremely detailed. In fact the spec has to define everything about the program, otherwise, nothing can be proven automatically and mechanically. Now, if the spec does define everything about how the program is going to behave, then, lo and behold, it contains all the information necessary to generate the program! And now certain geeks go off to a very dark place where they start thinking about automatically compiling specs into programs, and they start to think that they've just invented a way to program computers without programming.

--Joel Spolsky



Sometimes unit testing feels like testing 100 trees and hoping you have a forest...

Unit testing is sooooo
reductionist- but lots
of bugs happen in
interactions - in
between the parts -

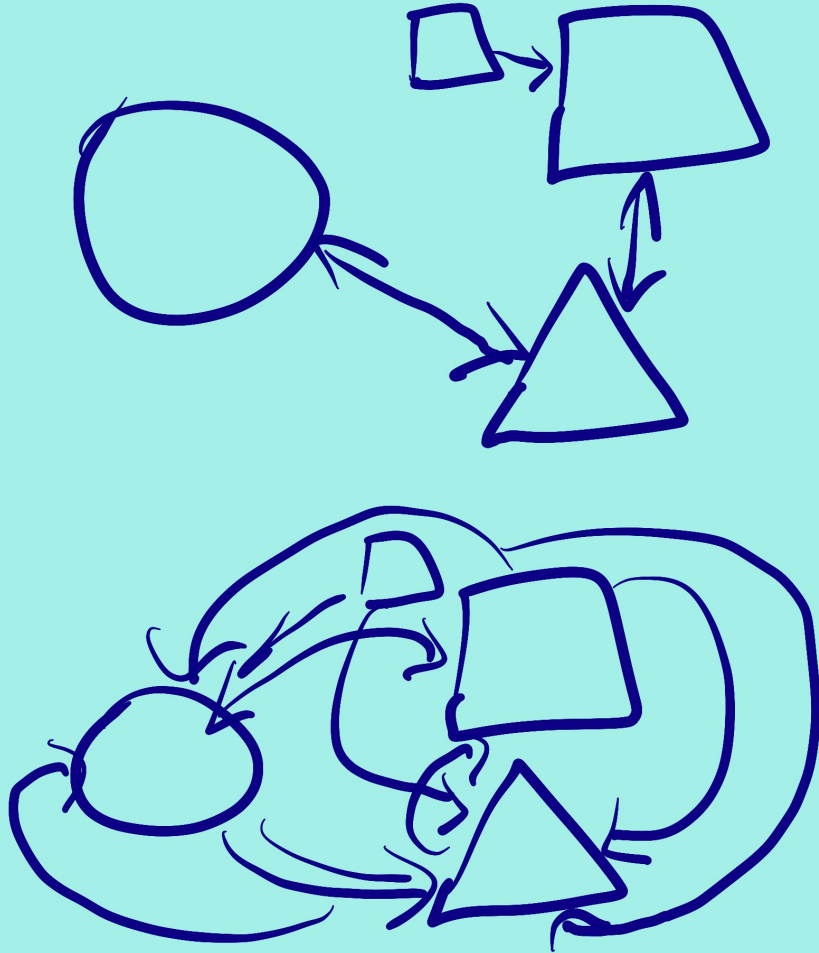




also, it's REALLY
hard to get coders to
search hard for
something they don't
want to see! Namely
bugs in their own
code...

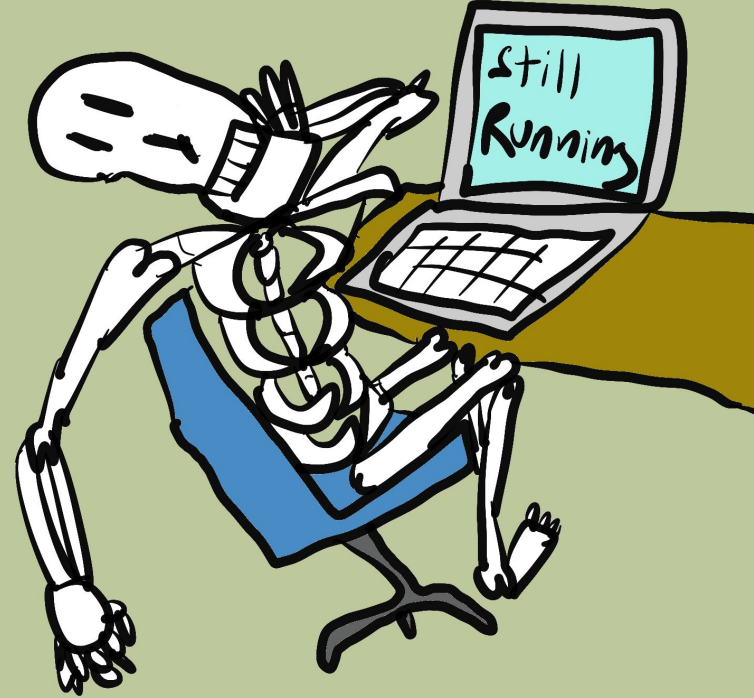
(also it's easier to test
"pure" functions with
just input and output,
no side effects-
unfortunately, UI
code is ALL ABOUT
the "side effects")





still, unit tests are
good to restructure
code so it's less
coupled

Conceptually I prefer functional tests - but they can take so long to run!





So we use mocks! But
how get the data for
those?

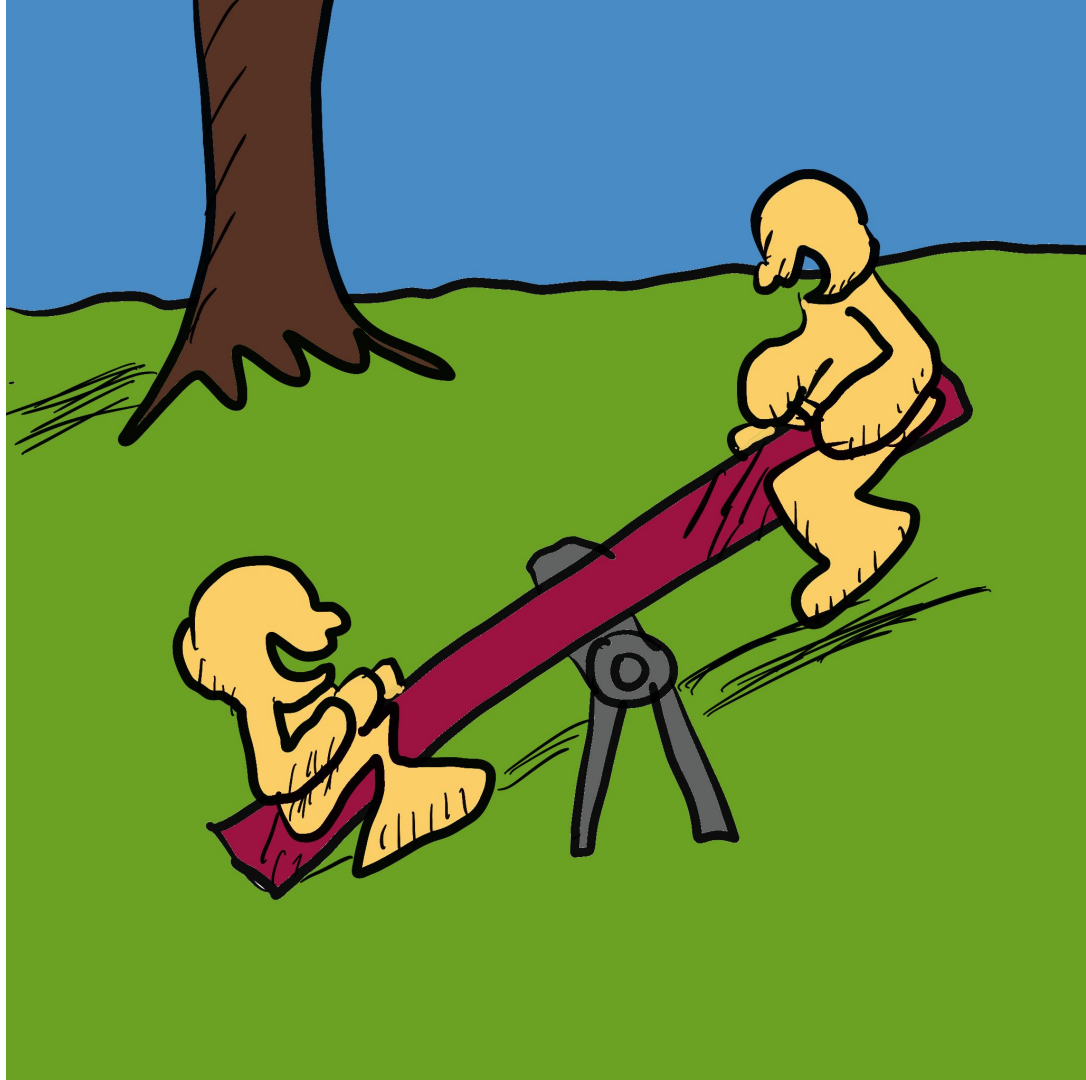
for ui stuff sometimes
we can grab it out of
the chrome inspector!

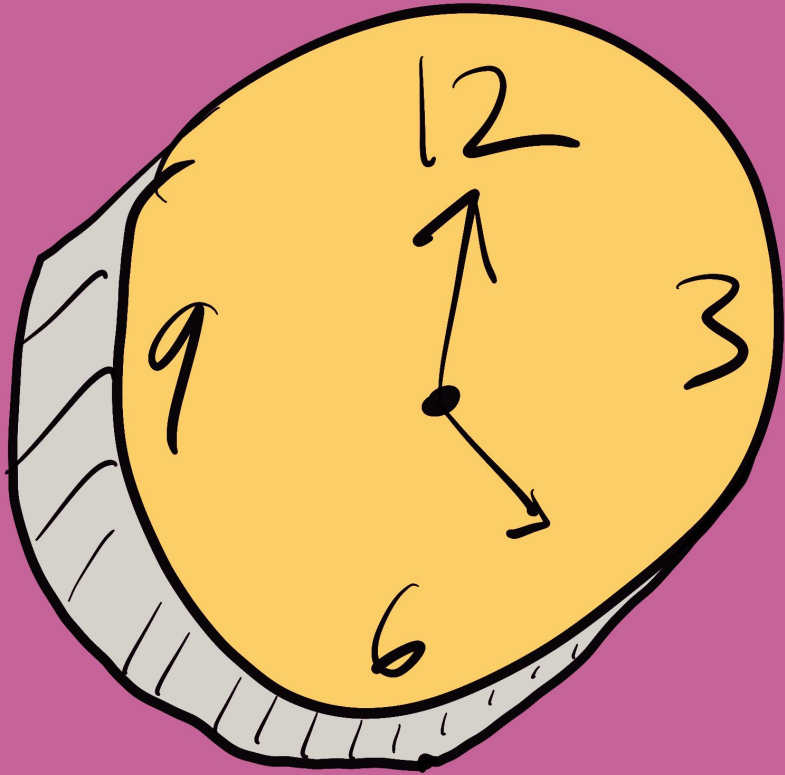
```
{ "ajax": "so much  
  "level": "high",  
  "reminder":  
  "elint doesn't  
  like raw JSON"
```



other time we can
build it up from
scratch

which is better? one is
'more real', the other
easier to understand

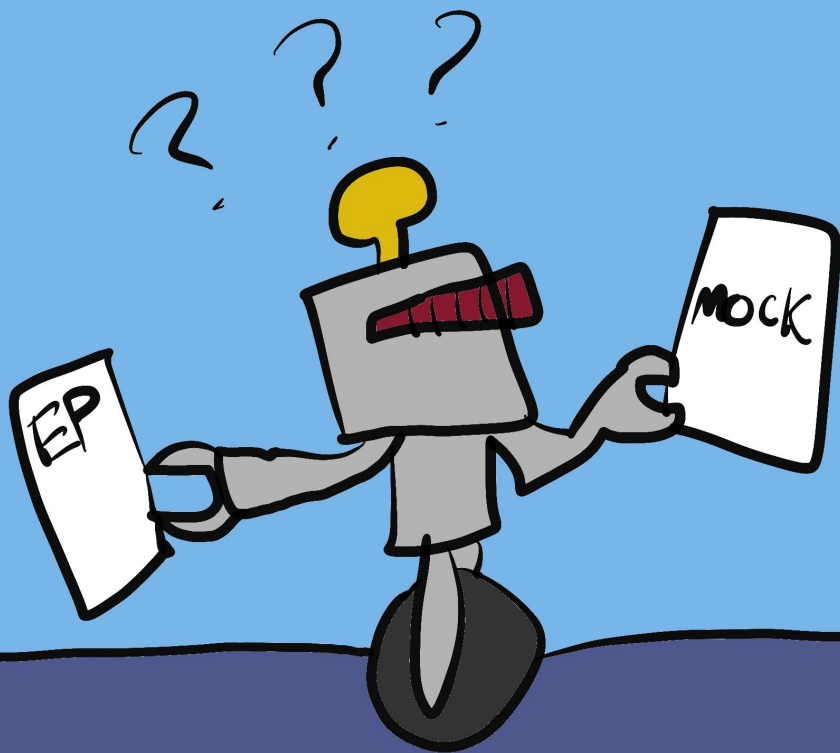




and what happens
over time? endpoint
data changes - that's
life!

and mocks tend to be
very fragile!
depending on the
signature to remain
the same is tough





so how can we
improve this process?
robots to make sure
the mocks line up with
the code?