

## David Evans - What developers and testers can learn from each other

Safe assumption: Code that's not tested doesn't work

People make mistakes...

An error is a difference in behaviour from what should have been beyond some margin of tolerance

↳ Things developers can help testers to learn

- ⇒ A defect report is just an opinion (until sb decides what action to do)
- ⇒ A defect is evidence of a missing test
  - A test is more expressive than a defect report

↳ End-to-end testing is overrated

- ⇒ Can only prove correctness of some paths in the system, but not the absence of errors
- ⇒ But of course you should have them!

↳ Acceptance TDD

- might slow down dev, but only to the extent as a bus is slowed down when passengers enter/leave at bus stops

↳ Exploring for value

- Giving test high value is important
- Amount of tests correlates with the importance we give the
- Tests are also specifications ⇒ helps when arguing with business people

↳ Coverage is always in reference to a model

- How good is your model? :-)

"All models are wrong. Some are useful."

⇒ Things that testers can help developers learn

↳ Testers are friends (not food)

↳ Why? before how?

Every test should start with an expectation

↳ write assertion first, then work backwards

Every change should start with a reason

↳ Reformulate your requirements:

As a ... <user> ⇒ In order to ... <solve problem/improve>

⇒ Good requirements do not presume solution design

→ don't exclude possible solutions by writing bad requirements