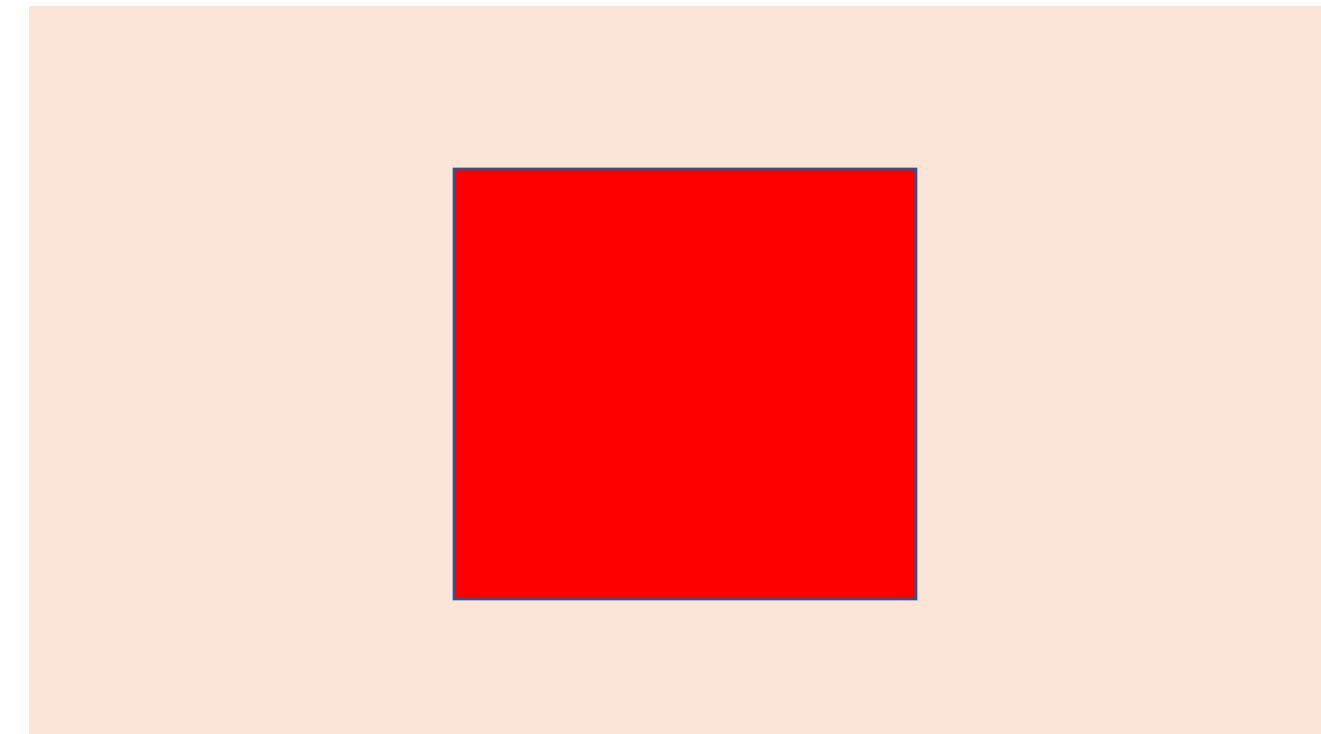
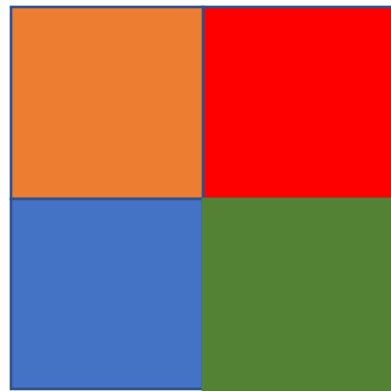


# Testing

**Write small programs to run parts of our code and verify the output**

**Unit tests**



**Integration tests**



# Testing



# Why test?

## Tests help preserving expected functionality

- As projects grow, it becomes easier to break things without noticing immediately
- Testing helps **detecting new errors early**
- Testing is essential for research software because we care about reproducibility of scientific results and because derivative research and programs depend on research software

## Tests help users of your code

- Make it easier for others to verify whether the code has been correctly installed
- Users of your code publish papers based on results produced by your code
- Your peers need to be able to reproduce your (to be) published computational results

## Tests help other developers

- Tests make it easier to **refactor the code** (rewrite code while keeping functionality)
- Documentation which is up to date by definition
- Easier for external developers to contribute to the project without breaking your code (**you may immediately see problems in your code but others may not**)

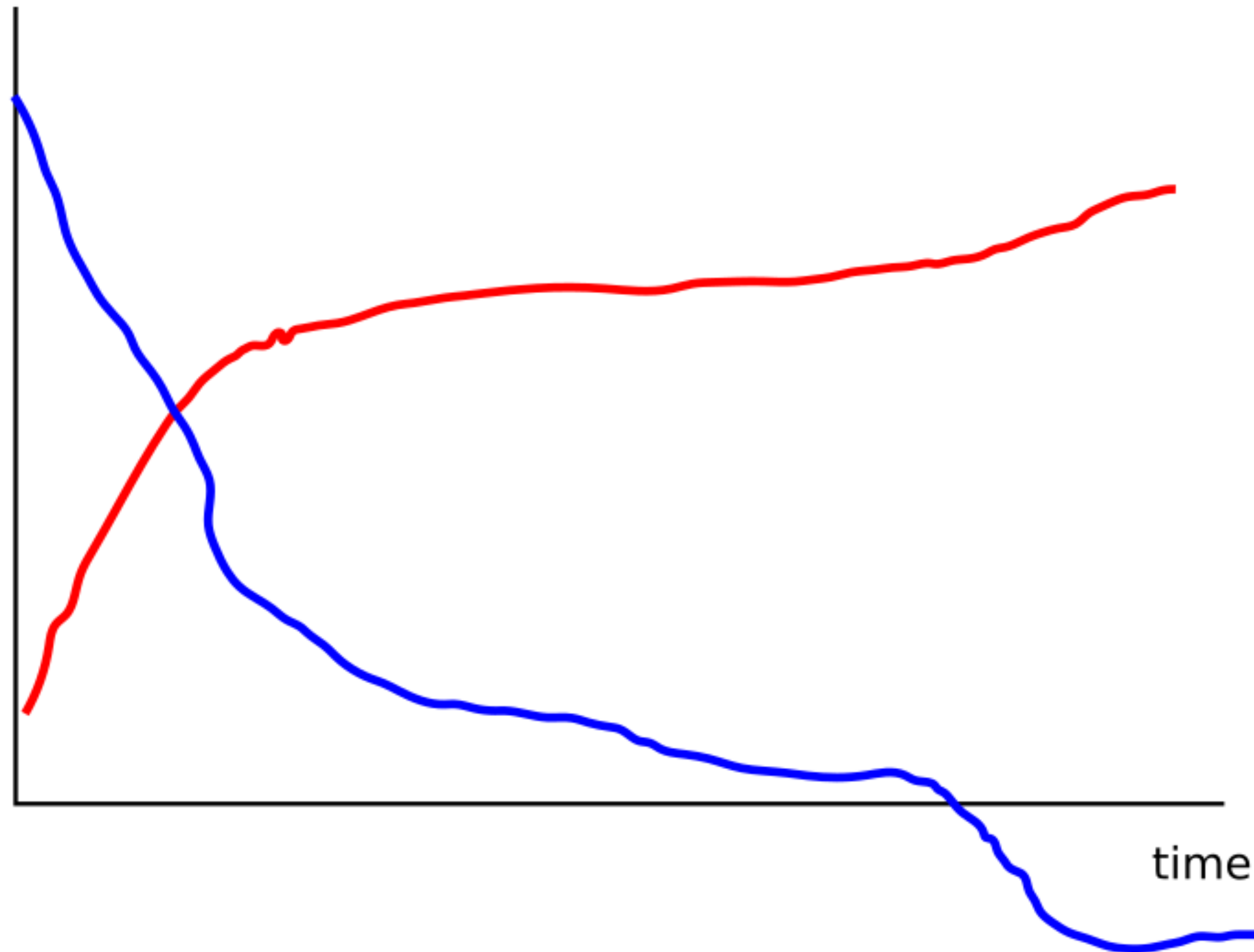
## Tests help managing complexity

**You write the test anyway!!  
and then throw it away**



# Why not test?

development speed



We are human

Code Review -> easier to tell someone else to write a test

Metrics like Codecov -> Stepping on the scale each morning

