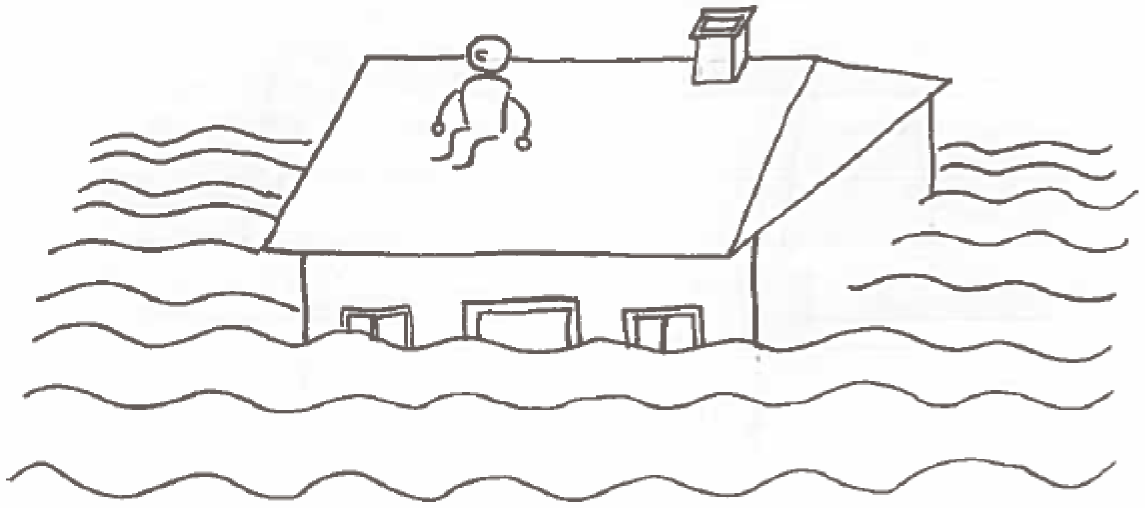
Python Best Practices

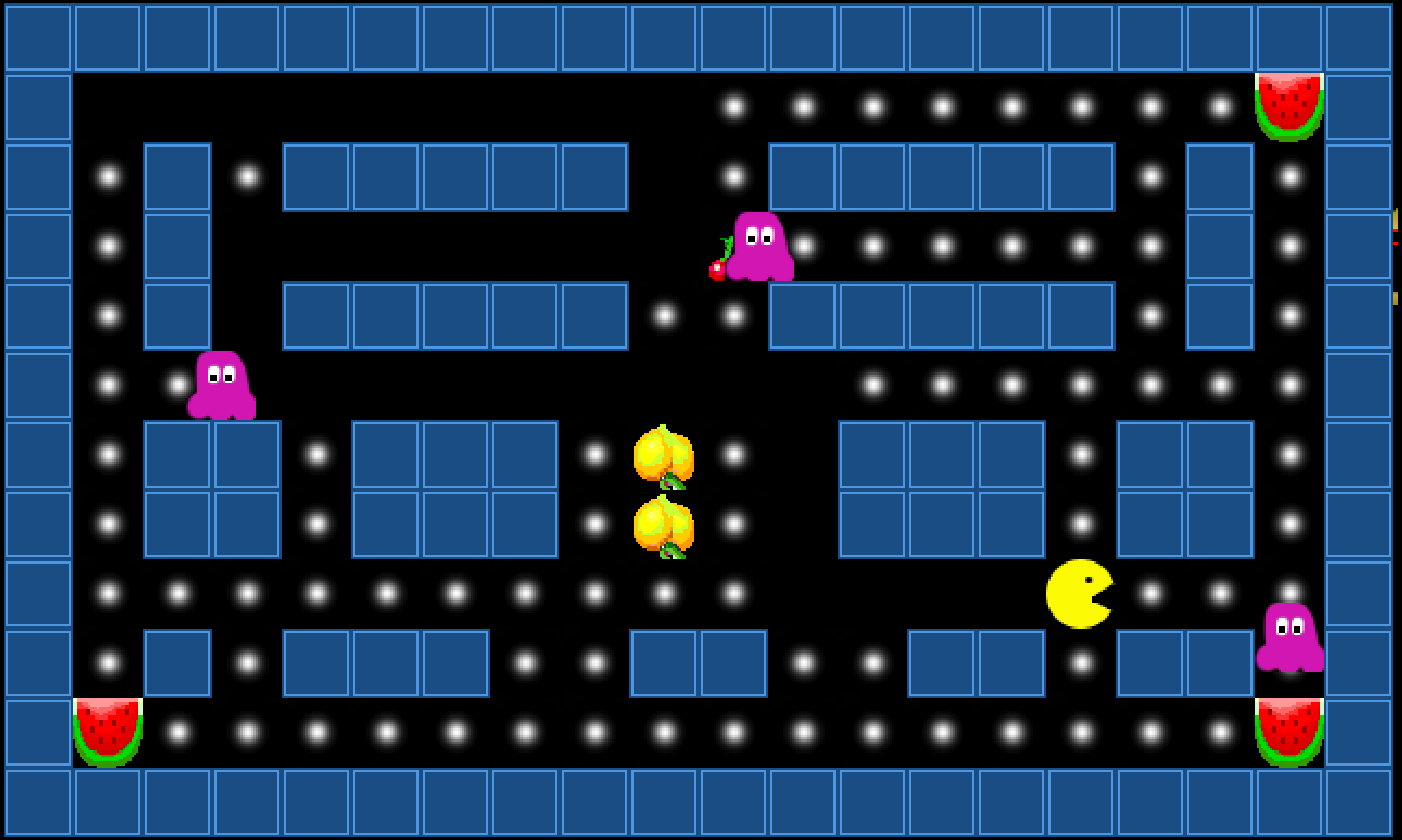
Dr. Kristian Rother



If Exceptions were a fire in the building,
this is what `except:pass` *looks like.*

ACADEMIS

krother@academis.eu
www.academis.eu



Source: www.youtube.com/watch?v=CayMeza487M

Source: ilesj.files.wordpress.com/2011/02/c64-and-easyflash.jpg

Merging two images

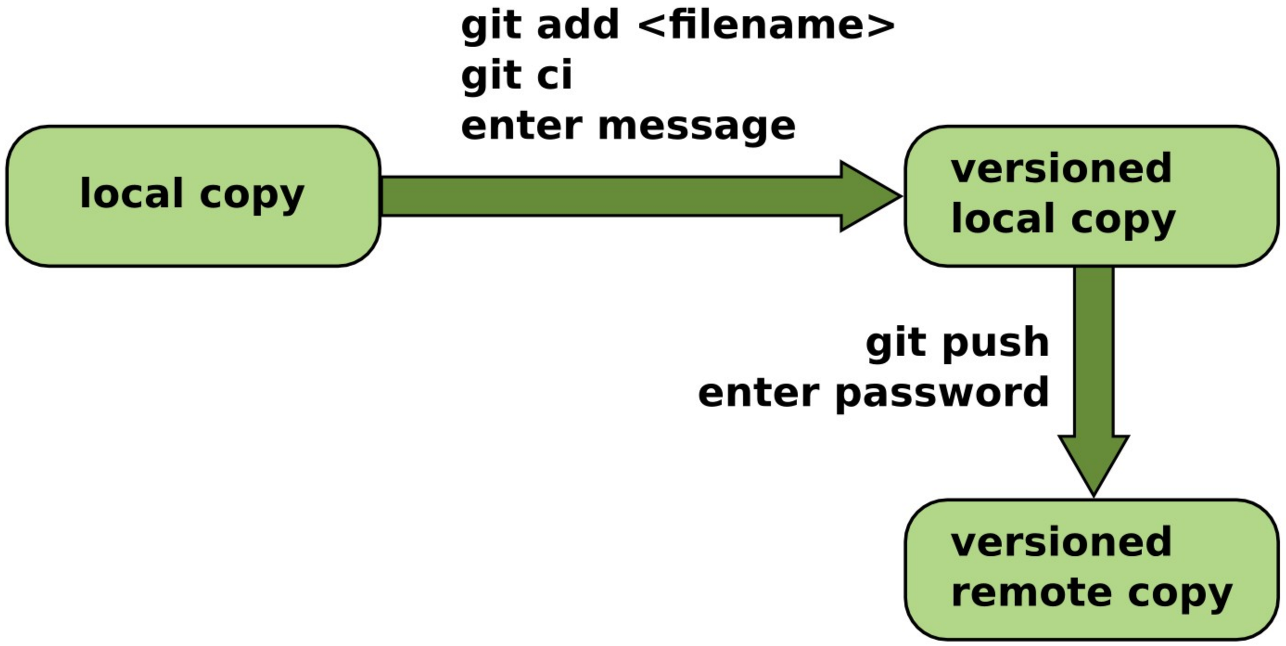
```
from pygame import image, Rect

maze = image.load('maze.png')
player = image.load('player.png')

maze.blit(player, Rect((32, 32, 64, 64)),
                  Rect((0, 0, 32, 32)))
image.save(maze, 'merged.png')
```

github.com/krother/maze_run

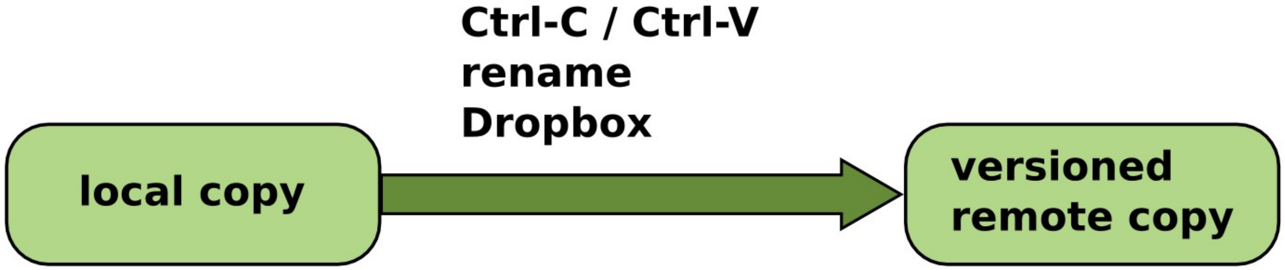
Python Best Practices

🟩 pdb	🟥 py.test	🟩 virtualenv
🟩 code reviews	🟩 TDD	🟩 Sphinx
🟥 git	🟩 continuous integration	🟩 setuptools
🟩 pylint	🟩 pyscaffold	🟥 ?

Version Control: Best Practice



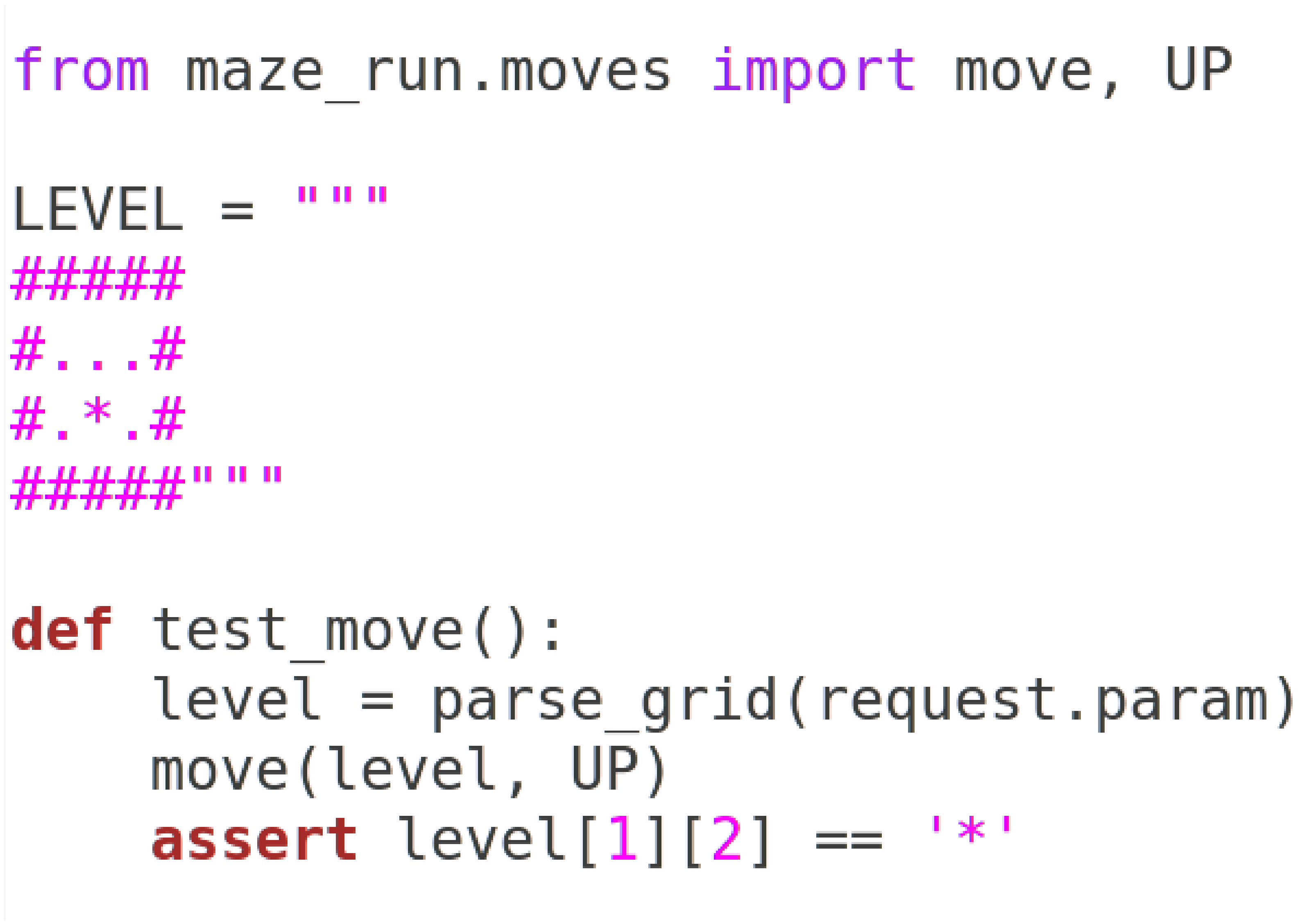
Version Control: Expect resistance!



Python Best Practices

pdb	py.test	virtualenv
code reviews	TDD	Sphinx
git	continuous integration	setuptools
pylint	pyscaffold	?

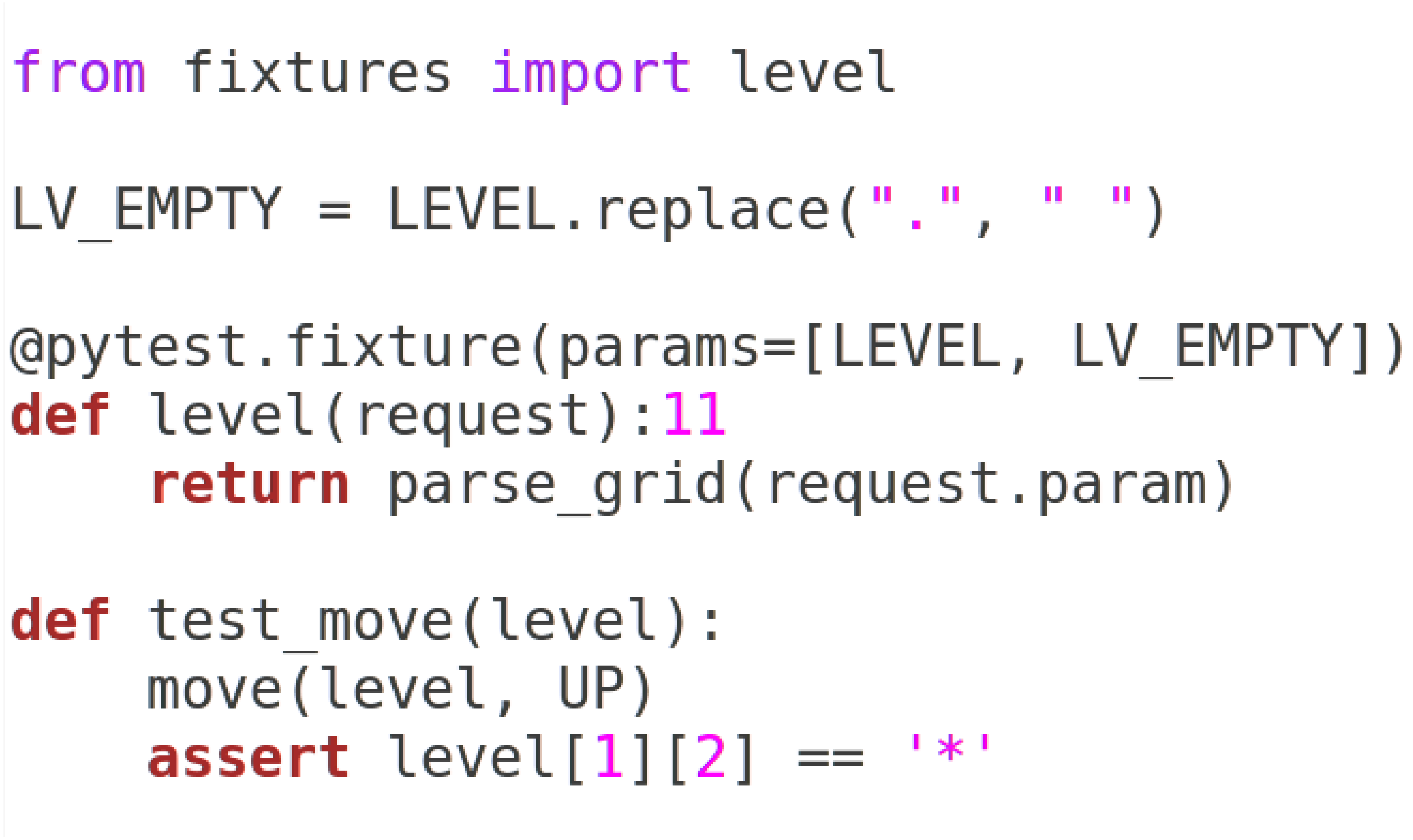
py.test



```
from maze_run.moves import move, UP

LEVEL = """
######
#...#
#.*.#
#####"""

def test_move():
    level = parse_grid(request.param)
    move(level, UP)
    assert level[1][2] == '*'
```

github.com/krother/maze_run

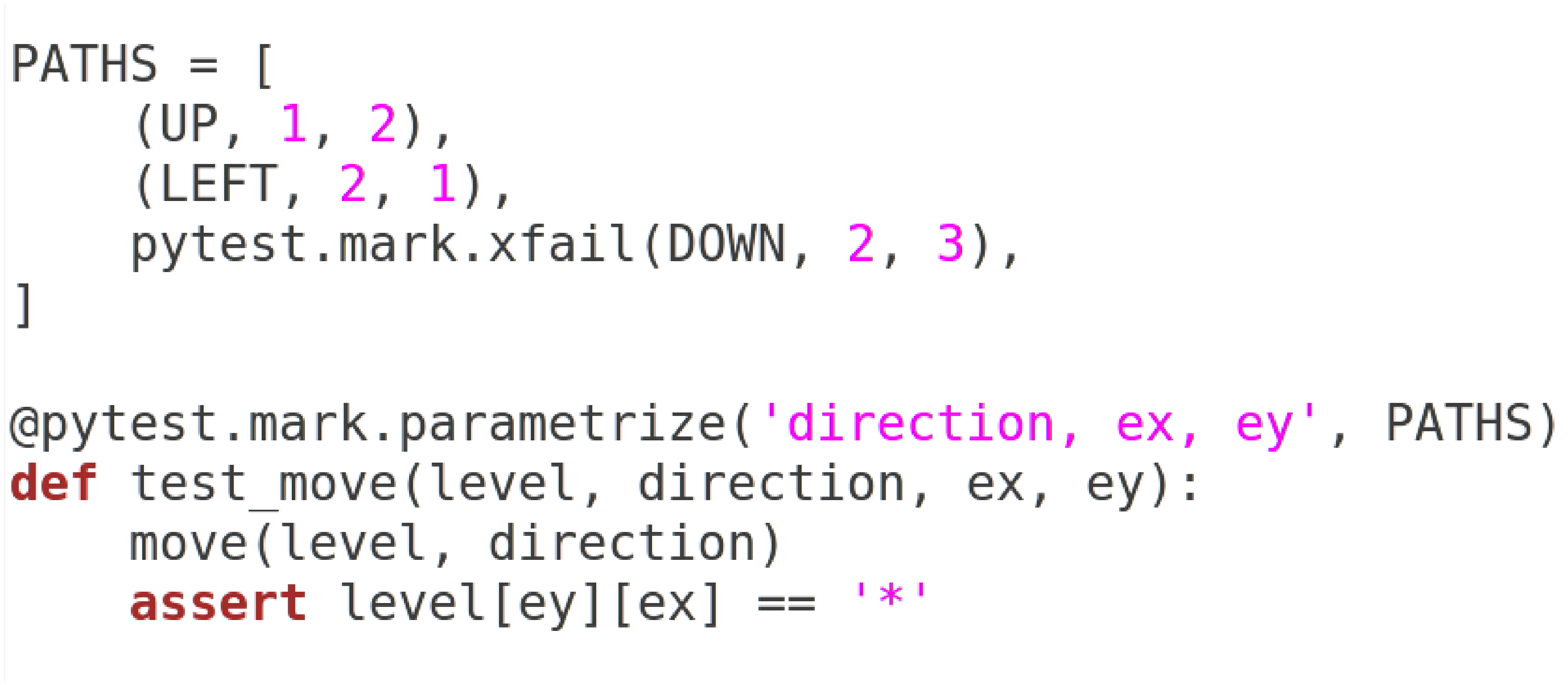
py.test fixtures



```python
from fixtures import level

LV_EMPTY = LEVEL.replace(".", " ")

@pytest.fixture(params=[LEVEL, LV_EMPTY])
def level(request):11
    return parse_grid(request.param)

def test_move(level):
    move(level, UP)
    assert level[1][2] == '*'
```

github.com/krother/maze_run

Parameterized tests



```python
PATHS = [
    (UP, 1, 2),
    (LEFT, 2, 1),
    pytest.mark.xfail(DOWN, 2, 3),
]

@pytest.mark.parametrize('direction, ex, ey', PATHS)
def test_move(level, direction, ex, ey):
    move(level, direction)
    assert level[ey][ex] == '*'
```

github.com/krother/maze_run

Source: www.youtube.com/watch?v=CayMeza487M

Python Best Practices

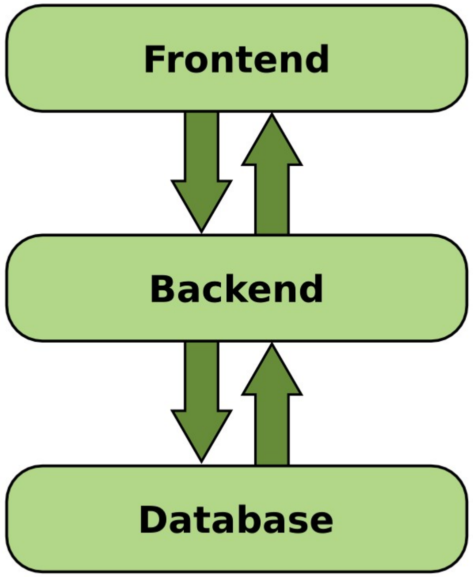
pdb	py.test	virtualenv
code reviews	TDD	Sphinx
git	continuous integration	setuptools
pylint	pyscaffold	?

Conway's Law

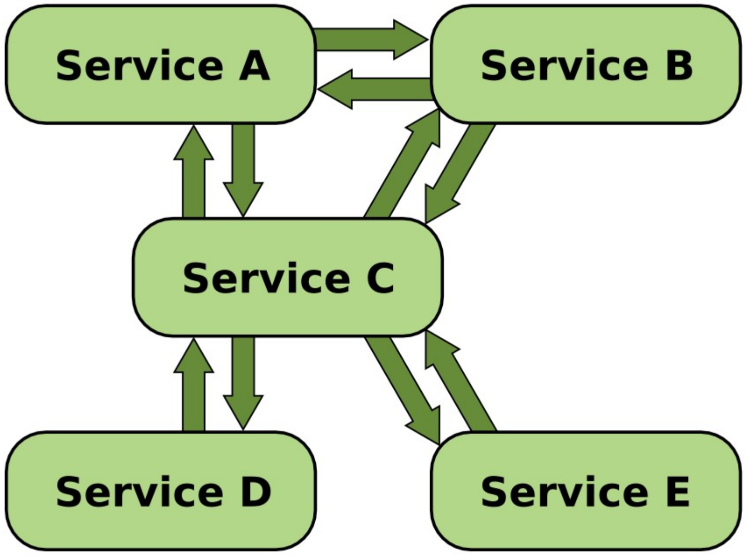
Any organization that designs a system
(defined broadly) will produce a design
whose structure is a copy
of the organization's communication structure.

www.melconway.com/Home/Conways_Law.html

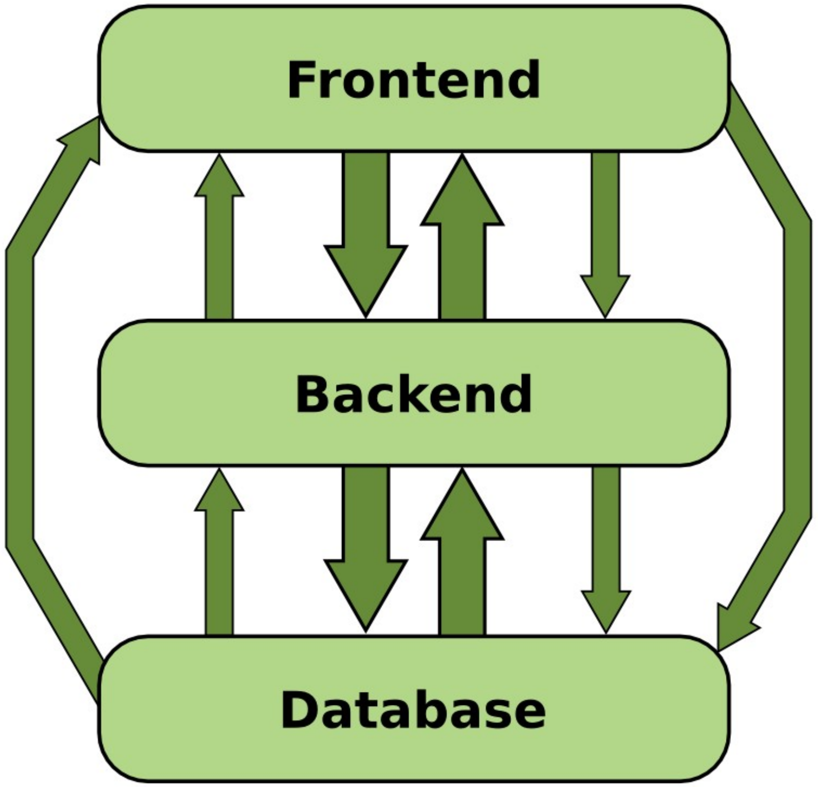
Layered teams



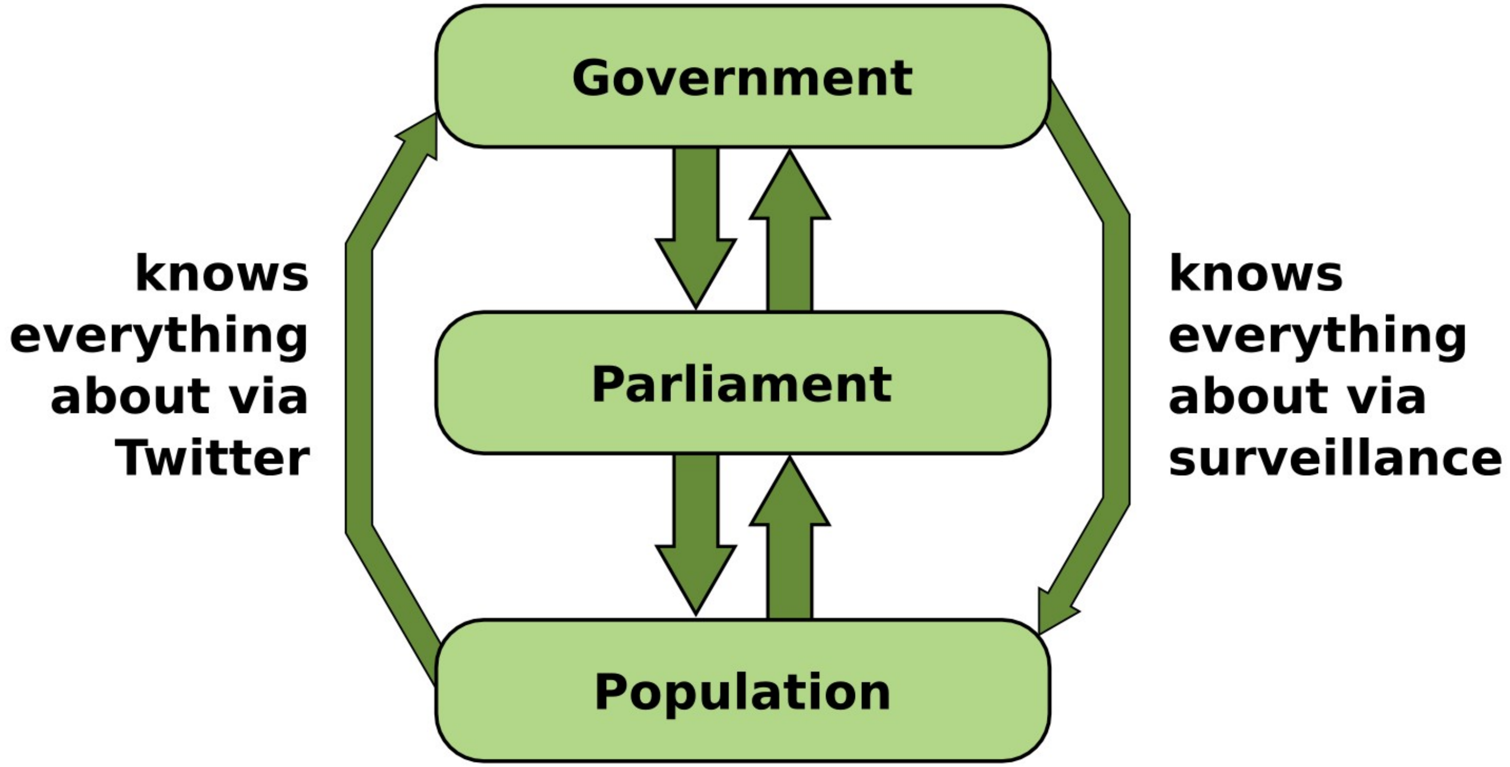
Small multifunctional teams



Colocaliziation without refactoring ..

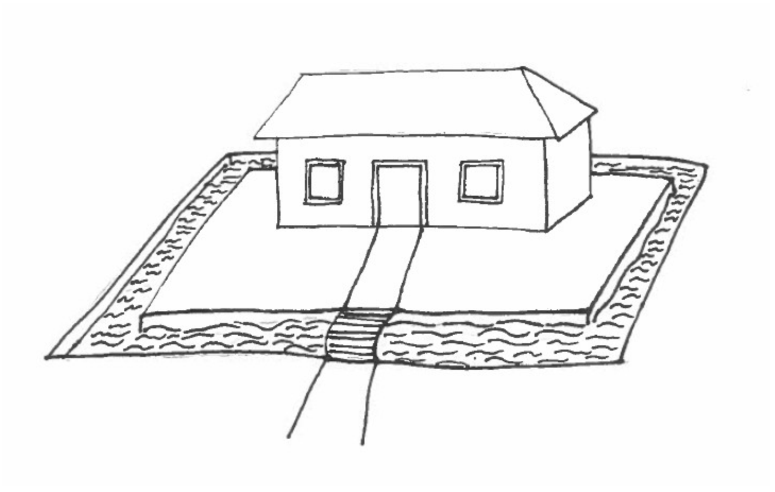


.. breaks the architecture..


Government
Parliament
Population
knows everything about via Twitter
knows everything about via surveillance

Best Practices

- an “invisible skill”.
- expect resistance
- advise one path
  to beginners.
- improve your toolset.



virtualenv creates a moat

around your software.

Python Best Practices

pdb	py.test	virtualenv
code reviews	TDD	Sphinx
git	continuous integration	setuptools
pylint	pyscaffold	?