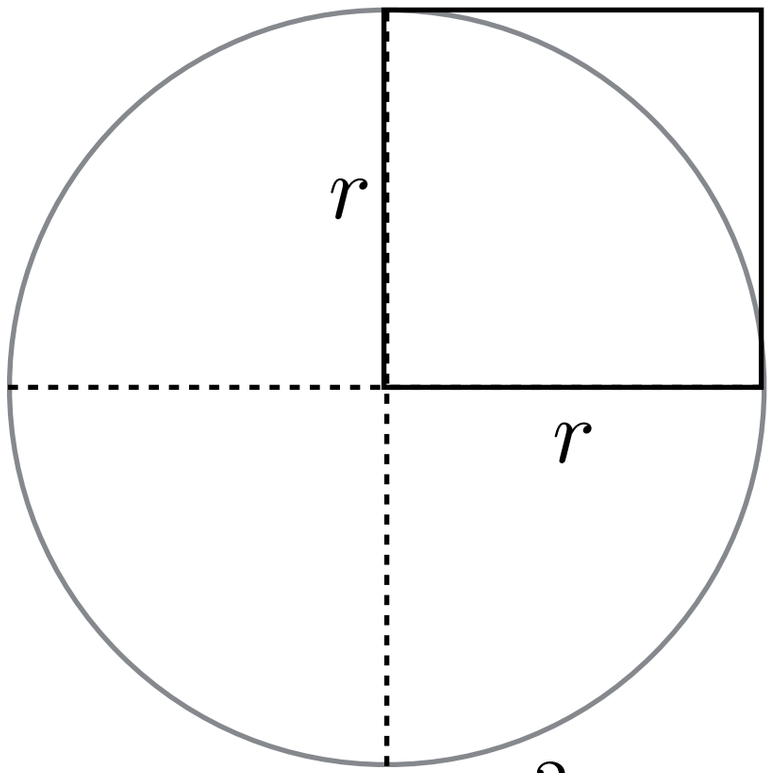
Freie Universität Berlin

Computational Sciences Projektseminar

Github workflow & Markov chain MC

Repetition: approximate π via direct MC

$$A_\square = r^2$$



$$A_\circ = \pi r^2$$

$$\frac{\frac{1}{4}A_\circ}{A_\square} = \frac{\frac{1}{4}\pi r^2}{r^2} = \frac{\pi}{4}$$

$$\chi(x,y) = \begin{cases} 1, & x^2 + y^2 \leq r^2 \\ 0, & else \end{cases}$$

$$\frac{\frac{1}{4}A_\circ}{A_\square} \approx \frac{1}{N}\sum_{n=0}^{N-1} \chi(x_n, y_n)$$

$$(x_n, y_n) \in [0, r]^2 \; \forall n$$

Repetition: approximate π via direct MC

```
def sample_pi(n):
    r = np.linalg.norm(np.random.rand(n, 2), axis=1)
    return 4.0 * np.sum((r <= 1.0)) / float(n)
```

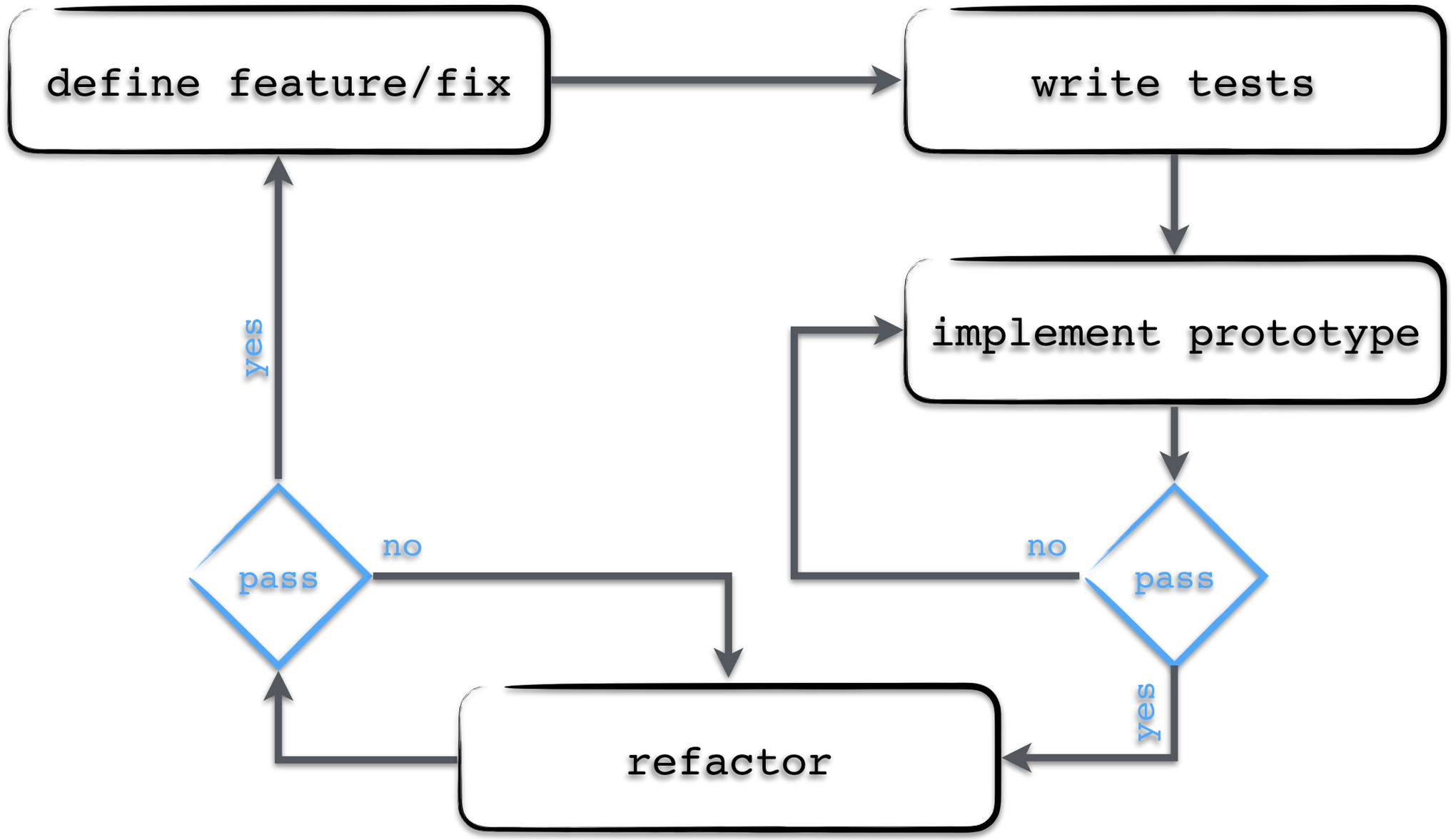
```
def sample_dist(n, m=100):
    return np.asarray(
        [sample_pi(n) for i in range(m)],
        dtype=np.float64)
```

https://github.com/markovmodel/compsci-2016

Exercise: github workflow

- team up with a colleague
- Student A sets up a repository with buggy code
- Student B forks the repository and creates a pull request (PR) with the bugfix
- Student A inspects/merges the PR

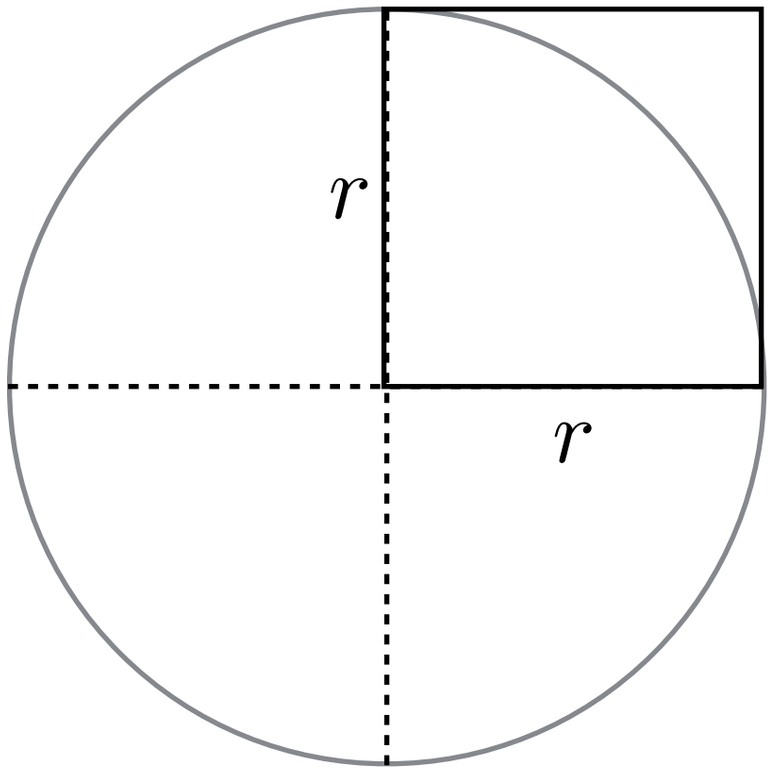
Test-driven development (TDD)



Test-driven development — why bother?

- a lot of work
- trusted units of code from the beginning
- automised testing (continuous integration, CI)
- easy to track bugs

Exercise: approximate π via Metropolis MC



$$\pi_{\text{sampled}} = \frac{4}{N} \sum_{n=0}^{N-1} \chi(x_n, y_n)$$

$$\mathbb{A}(x,y) = \begin{cases} 1, & (x,y) \in [0,1]^2 \\ 0, & else \end{cases}$$

$$\begin{aligned} (x_{\text{trial}}, y_{\text{trial}}) &= (x_{n-1}, y_{n-1}) + (\delta x, \delta y) \\ (x_n, y_n) &= \begin{cases} (x_{\text{trial}}, y_{\text{trial}}), & \mathbb{A}(x_{\text{trial}}, y_{\text{trial}}) = 1 \\ (x_{n-1}, y_{n-1}), & \text{else} \end{cases} \end{aligned}$$