


Crafting a robust deployment pipeline in finance

Boyan Mihaylov

Software architect and developer

@boyanio
<https://boyan.io>

Establish in-house development at a pension fund

A low-angle, upward-looking photograph of several tall skyscrapers against a clear blue sky. The buildings are arranged in a circular pattern, creating a sense of height and scale. The lighting suggests it might be late afternoon or early morning, with some windows reflecting the sky and others glowing with interior lights.

How does it feel inside
a financial institution?

Regulations & compliance

No errors allowed

Long decision-making process



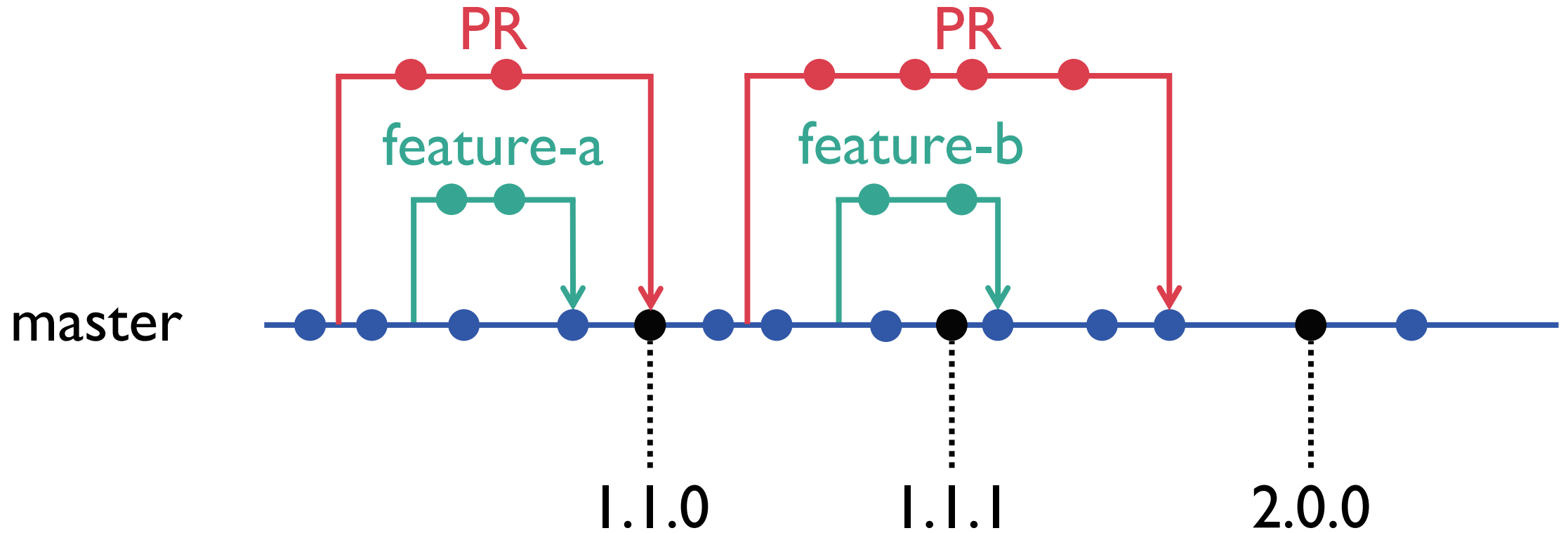
Goal: delivering **value** to our
members on a **frequent basis**

How does one create
a development process
from the scratch?

Start simple,
stay simple
as long as possible



Trunk-based development





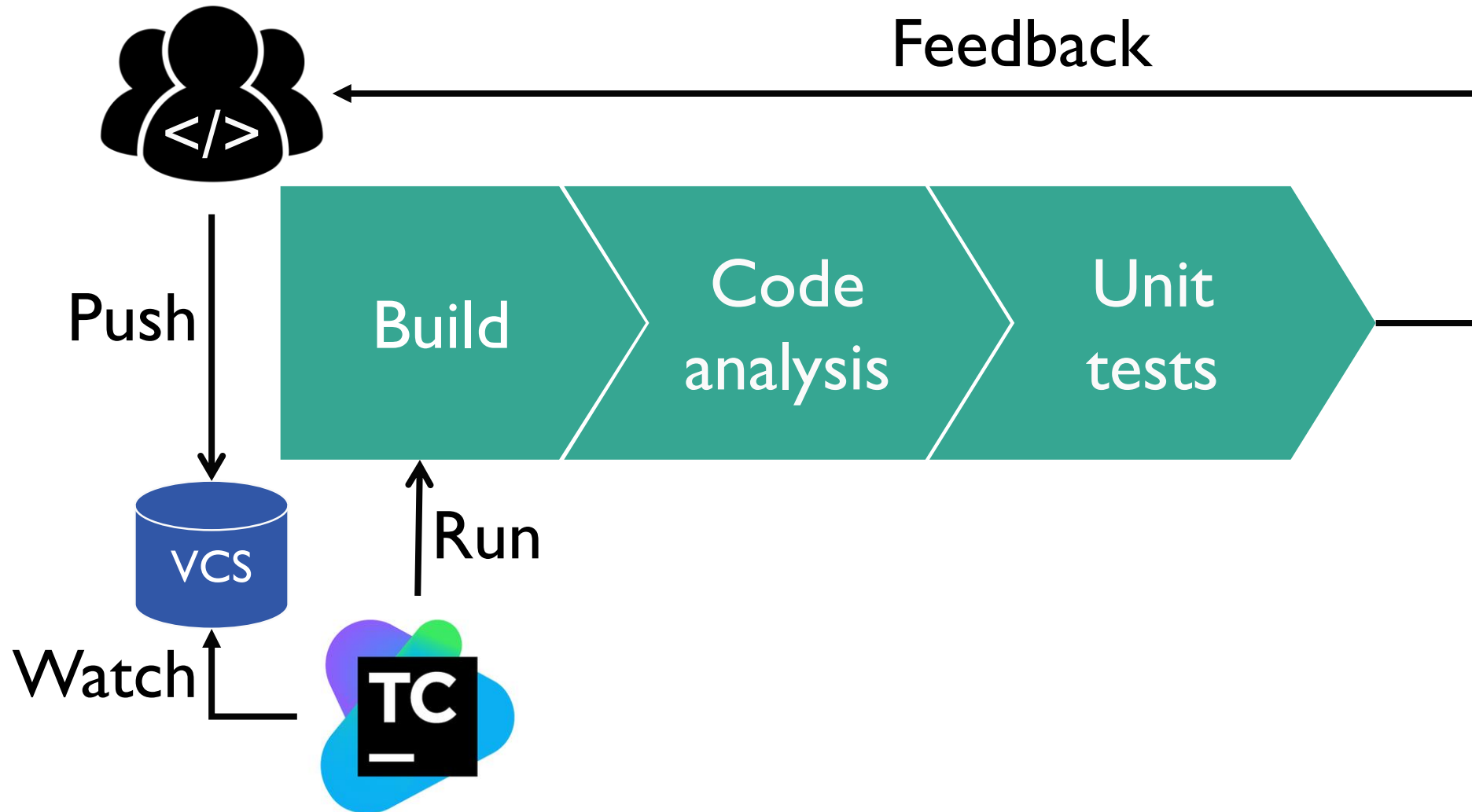
Long-lived branches disguise the big picture



**“Continuous integration (CI) is
the practice of merging all
developer working copies to a
shared mainline several
times a day”**

Wikipedia

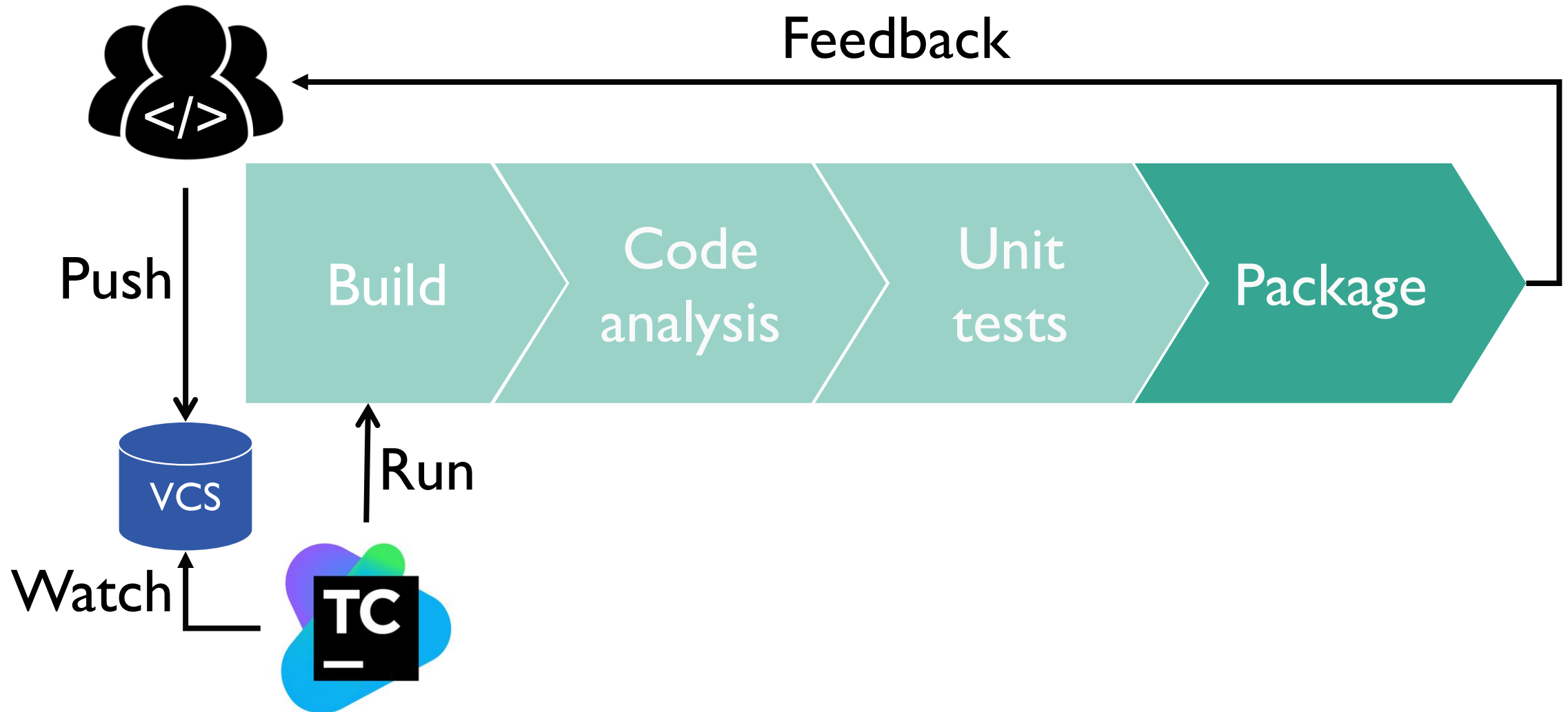
Continuous Integration



Always
deployable



Continuous Integration



Fact I.

Continuous integration
does not guarantee you
built the right thing

We want X



Let's build Y




Fact 2.

Continuous integration
does not guarantee you
built the thing right

```
public class BasketTests
{
    [Fact]
    public void ThrowsWhenNullIsPassedToAdd()
    {
        var basket = new Basket();

        Assert.Throws<ArgumentNullException>(
            () => basket.Add(null));
    }
}
```

 **auchenberg** / **volkswagen**

 Watch ▾

52

★ Star

5,986

 Fork

115

↔ Code

⚠ Issues **9**

🔗 Pull requests **38**

📁 Projects **0**

📖 Wiki

📊 Insights



Volkswagen detects when your tests are being run in a CI server, and makes them pass.

volkswagen

ci-server

cheat

defeat

javascript

nodejs

continuous-integration

🕒 94 commits

🔗 39 branches

🏷 15 releases

👤 12 contributors

⚖ MIT

Branch: master ▾

New pull request

Create new file

Upload files

Find file

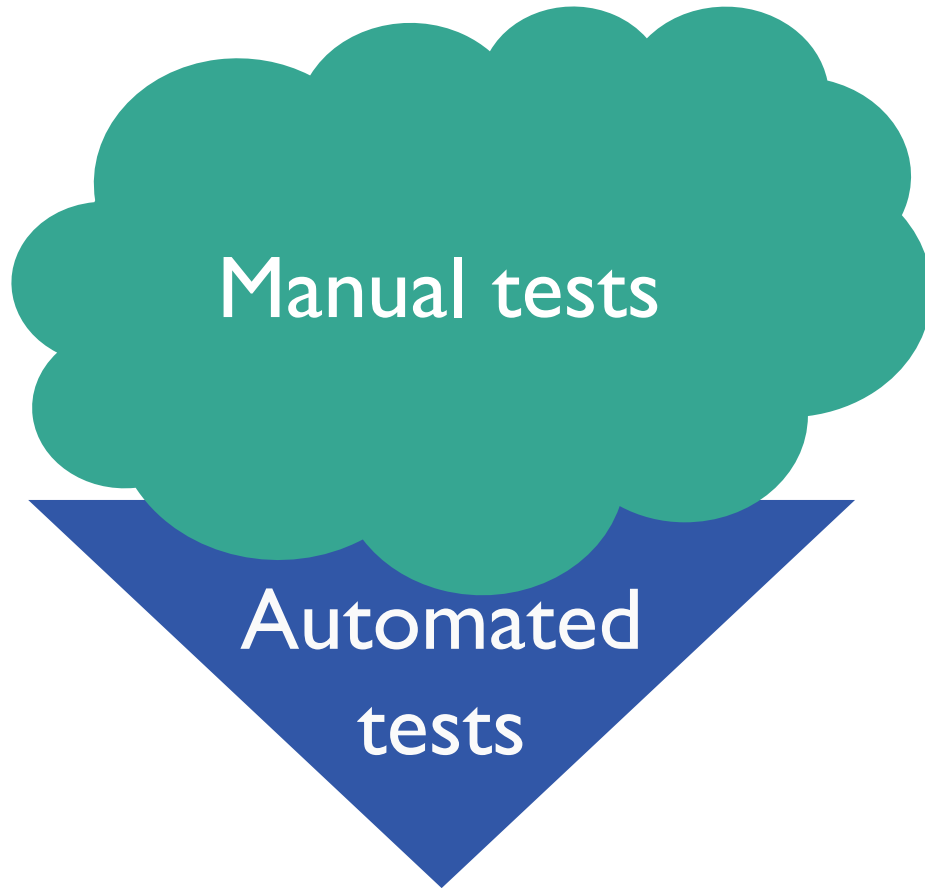
Clone or download ▾

Creating a development
process means creating
a development **culture**

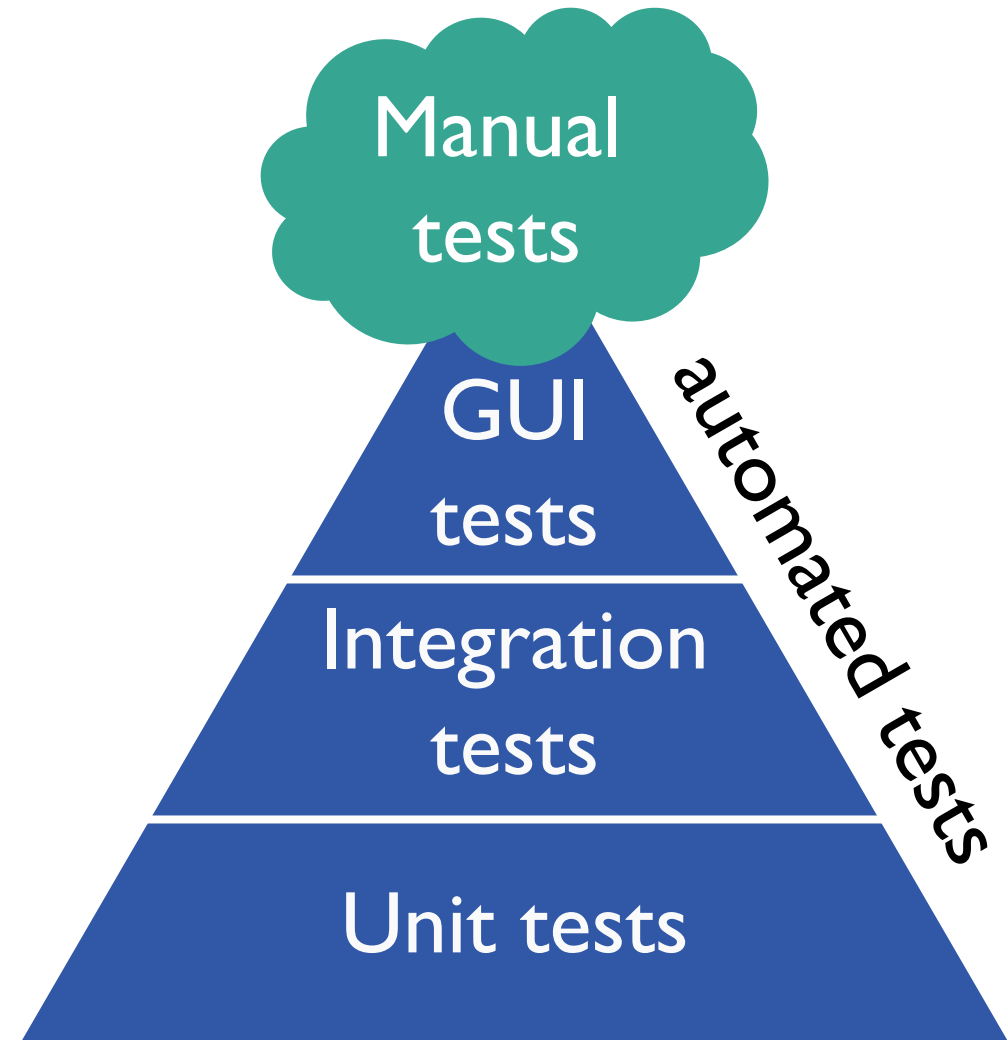


Building
trust





vs.



Steps to perform:

1. Go to the homepage
2. Log in as a member
3. Go to the retirement page
4. Go through the online retirement process
5. Sign your choices

Confirm:

- you got a confirmation e-mail
- there is a summary document in the archive
- there is a update request in the core system

Using Gherkin syntax

Feature: Retirement

Scenario: Complete retirement procedure

Given I am a member of the pension fund

And I am 67 years old

When I log in on the web portal

And I navigate to the retirement self-service

And I fill in my details

And I sign my choices

Then I should get a confirmation e-mail












And I should see a copy of my choices in my archive

And I should see my monthly pension payouts



pro•duc•tion [pruh-duhk-shuhn] (n)

I. This complex system of application code, distributed services, servers, networking gear, etc., upon which we are going to try to carefully apply a complicated set of changes and hope that nothing goes wrong. Cross your finders...here goes.

Name	Date modified	Type	Size
 _bin_old	12-Nov-13 13:03	File folder	
 bin	20-Dec-14 14:10	File folder	
 UI	12-Nov-13 13:03	File folder	
 Global.asax	12-Nov-13 13:03	ASP.NET Server A...	1 KB
 ReadMe-before-updates.txt	20-Dec-13 15:24	TXT File	11 KB
 ReadMe-remember-to-enable-log.txt	02-Jan-14 13:10	TXT File	11 KB
 Web.config	23-Jan-14 09:15	XML Configuratio...	6 KB
 Web-before-log.config	02-Jan-14 11:45	XML Configuratio...	6 KB
 Web-no-http.config	20-Dec-13 14:12	XML Configuratio...	6 KB
 Web-old - Copy.config	17-Nov-13 14:11	XML Configuratio...	6 KB
 Web-old.config	12-Nov-13 13:03	XML Configuratio...	6 KB



Manual deployment
is dangerous

Automated deployment to any environment



Identical environment set-up

Build once, deploy everywhere



Octopus Deploy

TM-E	TM-C	TM-D	PreProd	Production
				✓ 1.3.2 September 20th 2017
		✓ 1.1.6 October 4th 2017		✓ 1.1.3 July 26th 2017
	✓ 1.6.23 September 11th 2017	✓ 1.6.23 August 18th 2017	✓ 1.6.23 August 23rd 2017	✓ 1.6.23 August 23rd 2017
✗ 3.4.13 November 7th 2017	✓ 3.4.5 October 6th 2017	✗ 3.4.15 November 8th 2017	✓ 3.4.12 November 1st 2017	✓ 3.4.12 November 1st 2017
✓ 1.2.0 September 13th 2017	✓ 1.2.0 September 13th 2017	✓ 1.2.0 September 13th 2017	✓ 1.2.0 September 13th 2017	✓ 1.2.0 September 20th 2017
✓ 2.0.6 September 28th 2017		✓ 2.0.6 September 28th 2017		✓ 2.0.6 October 3rd 2017
	✓ 1.7.5 September 14th 2017	✓ 2.0.1 October 4th 2017	✓ 1.7.5 September 12th 2017	✓ 1.7.5 September 12th 2017
✗ 2.50.15 November 7th 2017	✓ 2.50.4 October 6th 2017	✗ 2.50.17 November 8th 2017	✓ 2.50.11 November 1st 2017	✓ 2.50.11 November 1st 2017
	✓ 2.0.0 March 2nd 2017	✓ 2.0.0 March 1st 2017	✓ 2.0.0 March 2nd 2017	✓ 2.0.0 March 2nd 2017
✗ 1.14.20 November 7th 2017	✓ 1.14.9 October 6th 2017	✗ 1.14.20 November 7th 2017	✓ 1.14.17 November 1st 2017	✓ 1.14.17 November 1st 2017
✓ 1.4.4 September 20th 2017				
	✓ 1.0.0.2 October 24th 2017	✓ 1.0.0.2 October 24th 2017		

Production deployment process

- Deploy to a fictive-data test environment
- Run automated UI tests
- Promote to an acceptance test environment
- Perform manual tests
- Create RFC
- Promote to production

How often shall we
deploy to production?

The fear of deployment



When you fear something,
you should do it often enough.
Then it becomes trivial.

Days between deploying to Production



Release

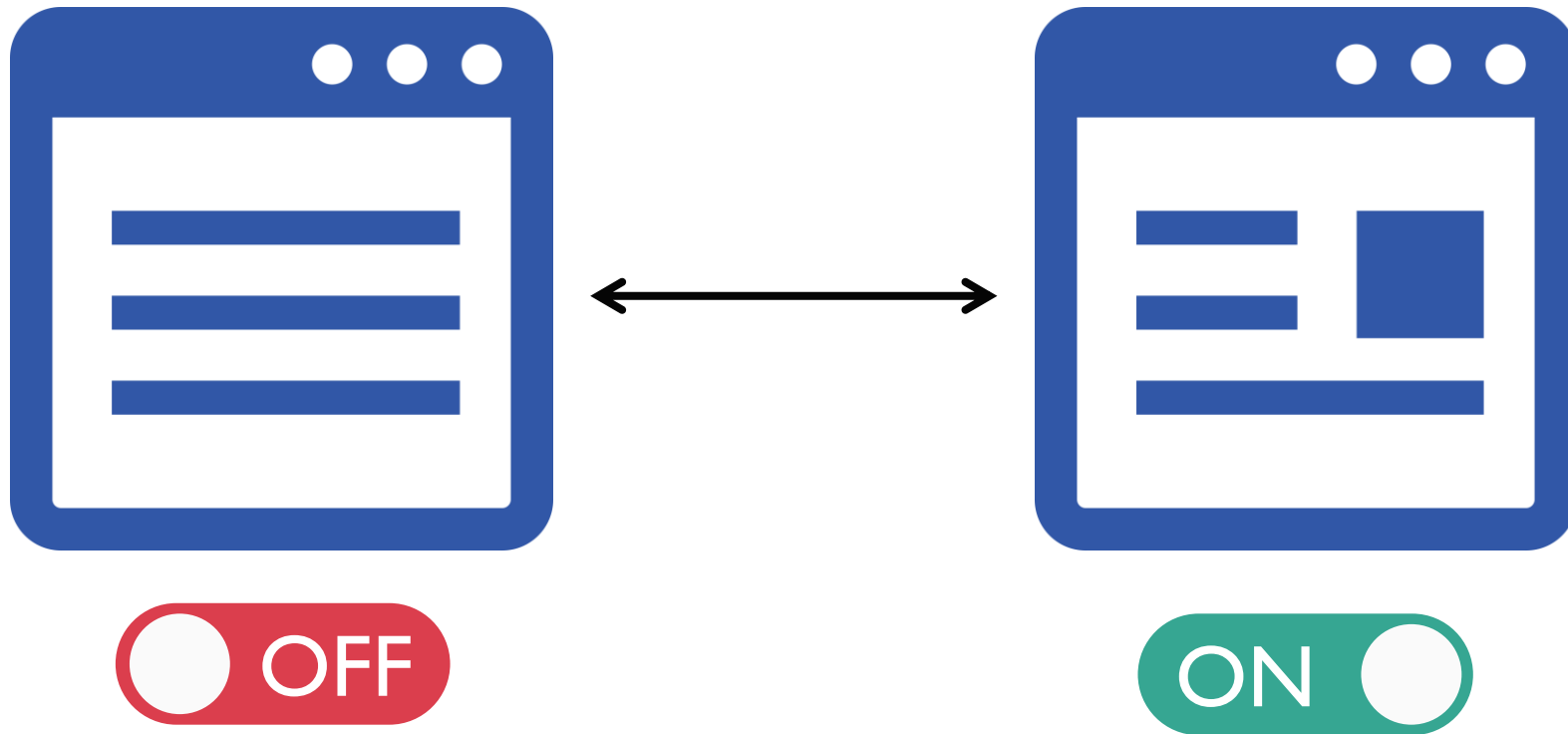
The act of making single or multiple features available to a wider audience

Deploy

The act of installing and configuring a software package onto a target system

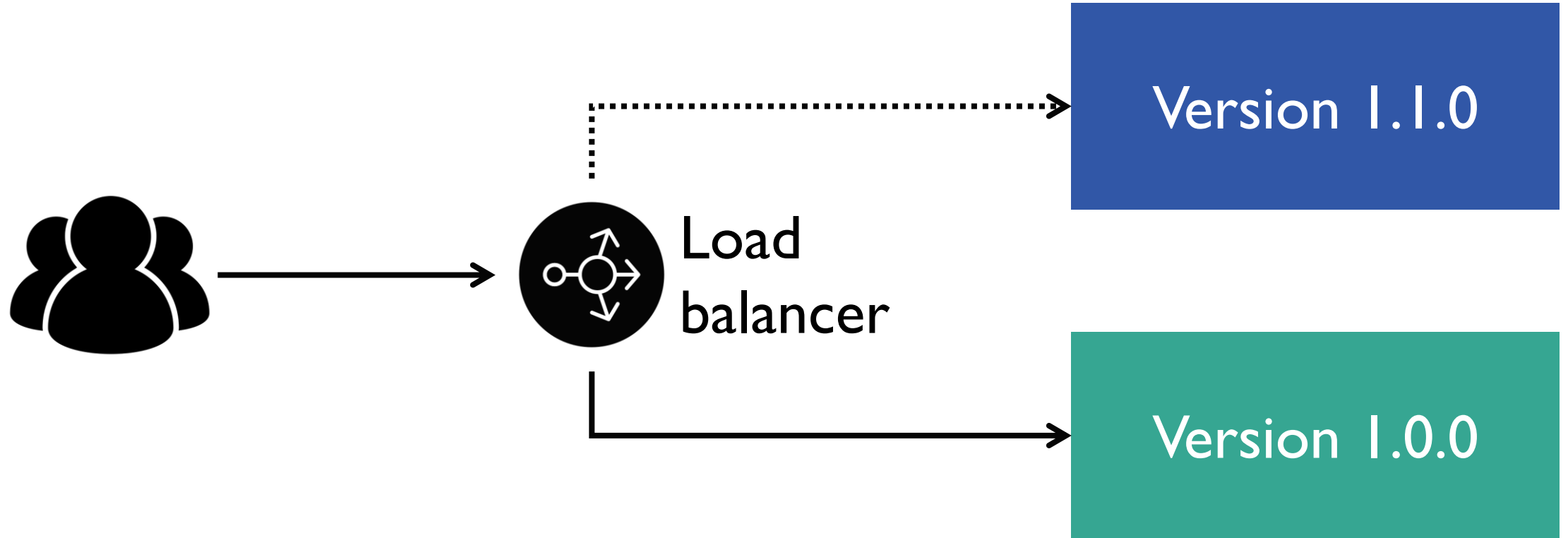
Feature toggles

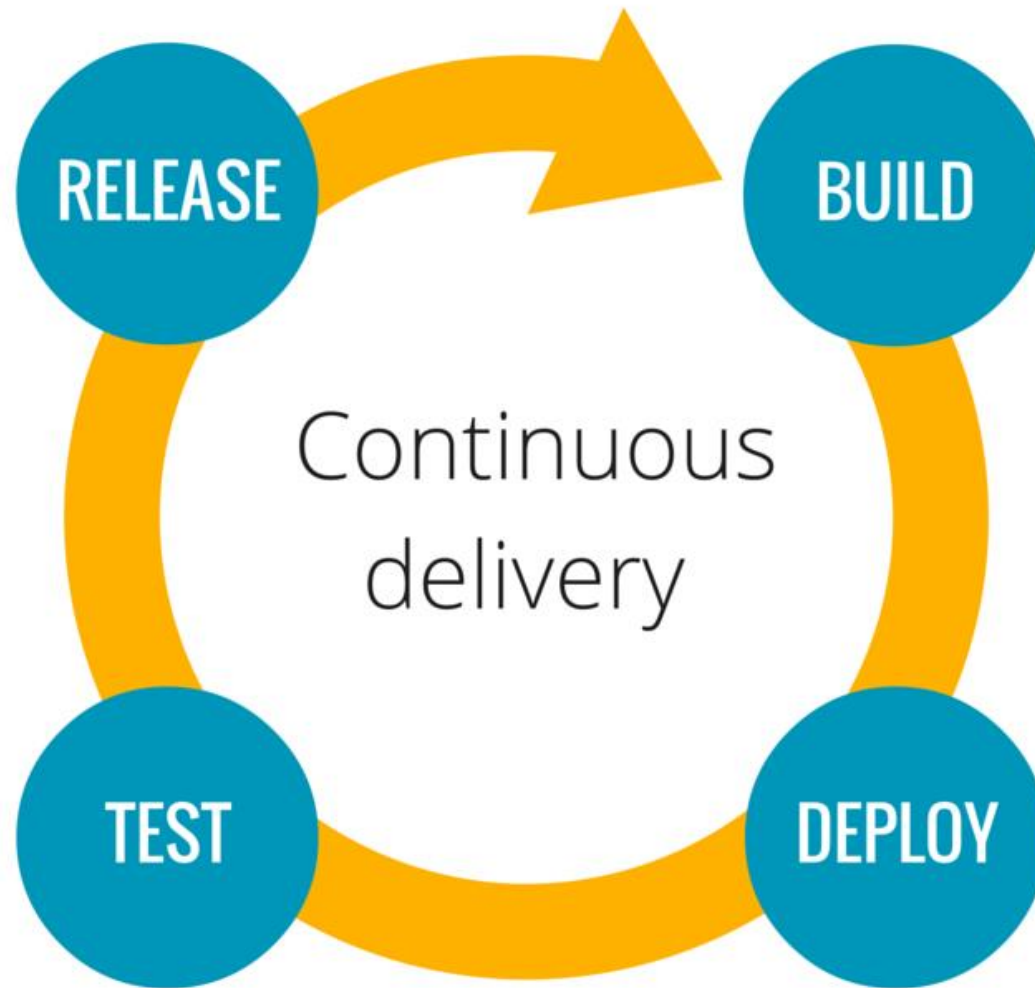
let you deploy often without releasing immediately



Zero downtime deployment

using blue-green deployment approach

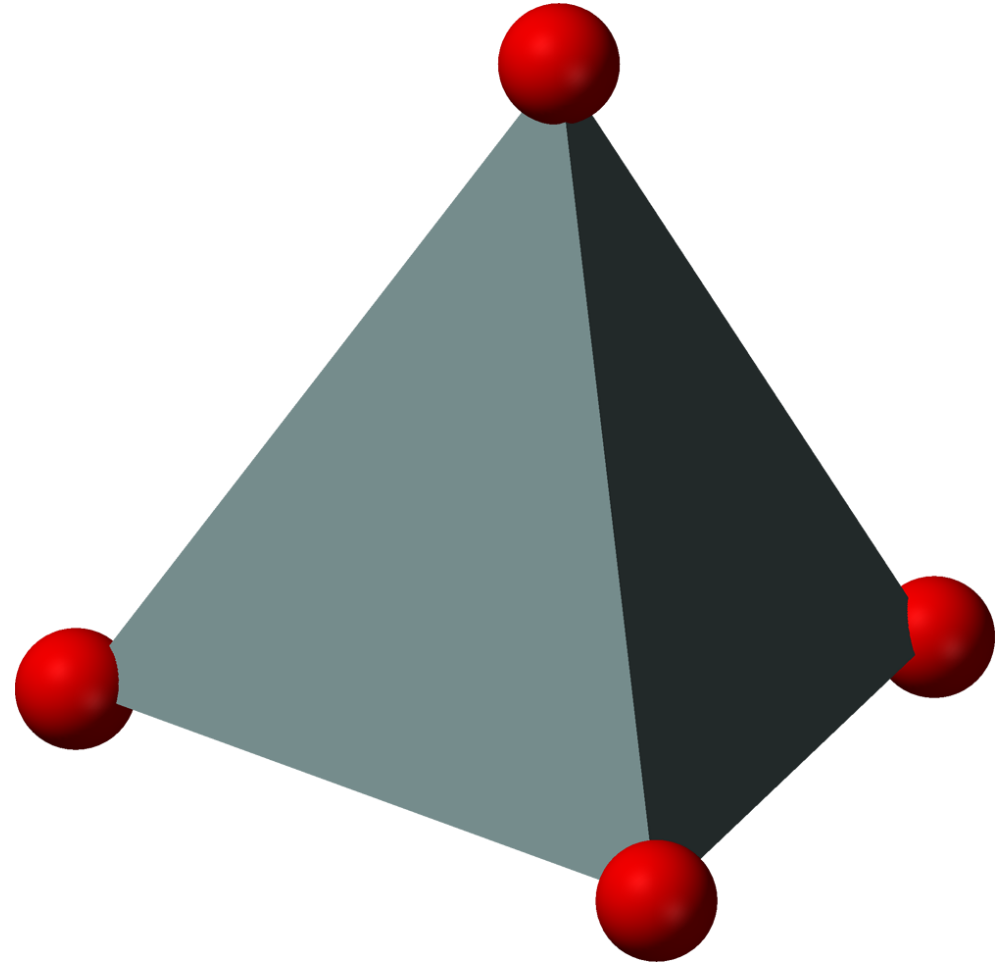




Q: Shall we move to
Continuous Deployment?

A: We can, but we don't have to

How robust is
the deployment
pipeline?





The deployment pipeline supports the established agile practices and desired goals

Keep things simple
Keep things identical
Build trust in your delivery
Automate the repeatable
Don't hype, make it work for you

Robust deployment pipeline
is about confidence in your
value delivery

Boyan Mihaylov

@boyanio

<https://boyan.io>