AGILE AND DEVOPS FOR DATA SCIENCE

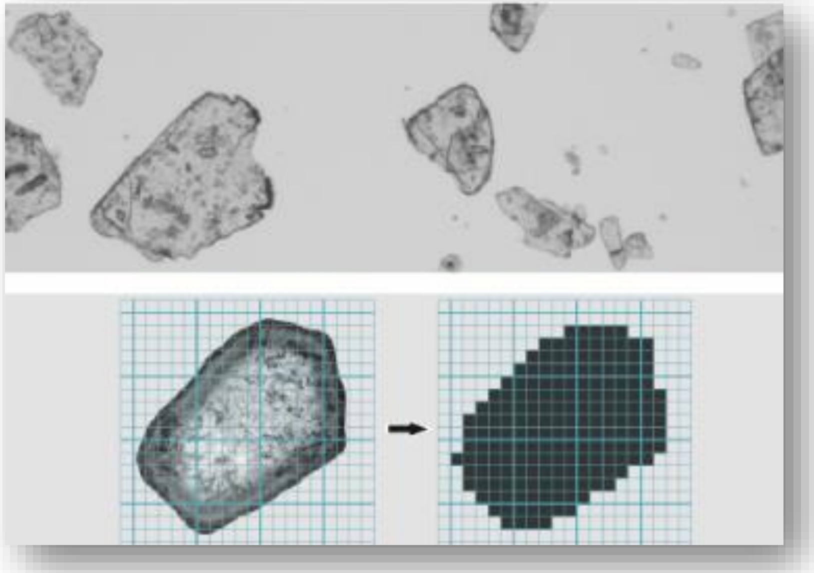
Chris Musselle

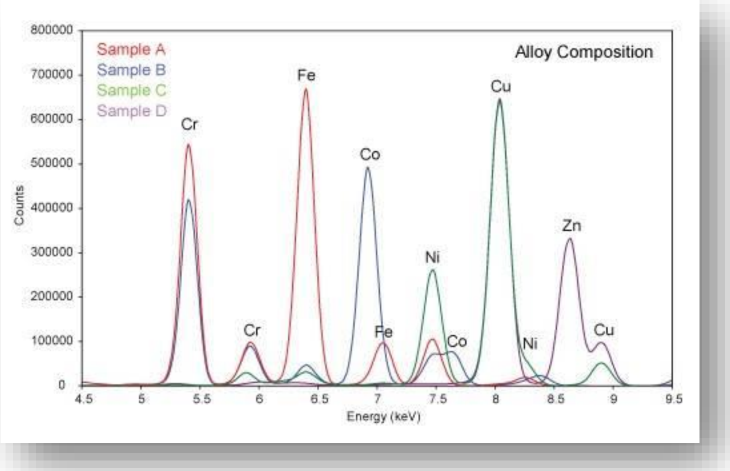
INTRODUCTION

Data Scientist at Malvern Panalytical

50% Data Scientist 50% Engineer

- R&D and Prototyping
- Productionising Data Science Projects



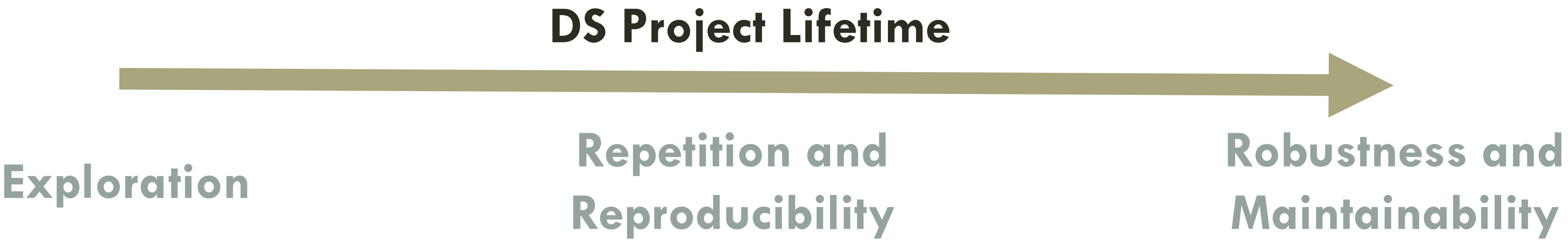


LIFE AS A DATA SCIENTIST

Involves a lot of programming, often a self-taught skill.

A mixture of exploratory R&D work, prototyping and communication.

Working on many projects, lots to learn all the time!



DS Project Lifetime

Exploration

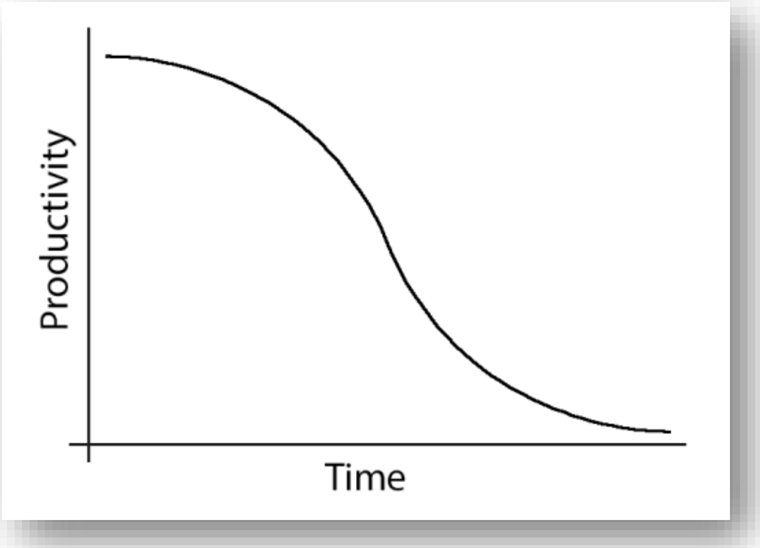
Repetition and Reproducibility

Robustness and Maintainability

HOWEVER…

For our work to add value, people need to make use of it. Leading to…

- Requests for more features,
- Bugs to fix
- Changes to requirements
- Issues with new data



Programming for robustness and maintainability is difficult.
But without it we lose more time with every change or fix.

Don’t want to spend whole time patching and fixing!
Software development shares these challenges too.

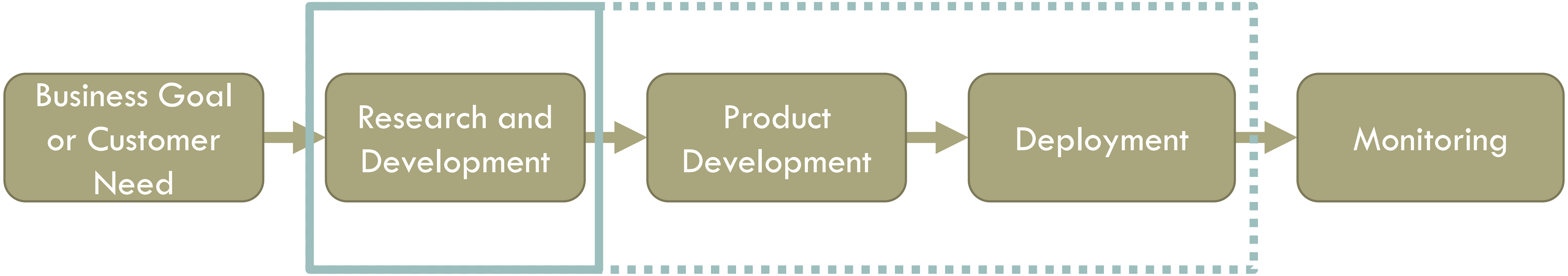
SO…

Can modern software development offer guidance here?   Yes!

- **Automated Testing** – Quickly proving the quality of the code
- **Agile Software Development** – Incremental and iterative delivery
- **DevOps** – Fast flow, Quick feedback, Continual Improvement

Not a quick fix, but more of a road map towards a better way!

DATA SCIENCE VALUE PIPELINE

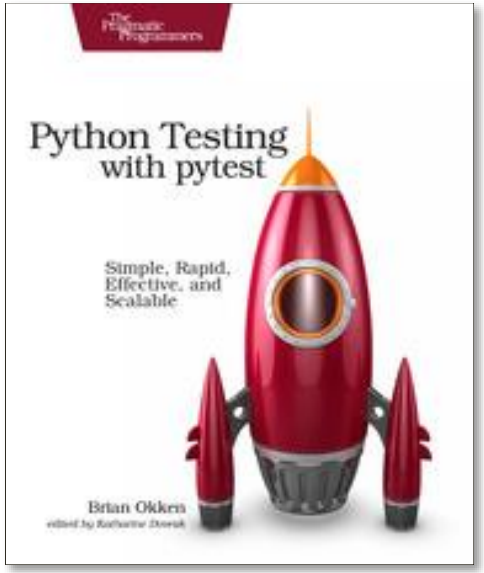


- The more we help and work with downstream steps, the more everyone wins.
- For our work to really add value, people need to make use of it.
- How can we achieve fast flow whilst also addressing robustness?

AUTOMATED TESTING

- Gives **assurance** that code works, and quick **feedback** when it doesn't!
- Allows you to make changes with more confidence
- Without checks in place, every change is increasingly risky

- Python Testing with pytest - Brian Okken


The Pragmatic Programmers
Python Testing with pytest
Simple, Rapid, Effective, and Scalable
Brian Okken
edited by Katharine Dvorak

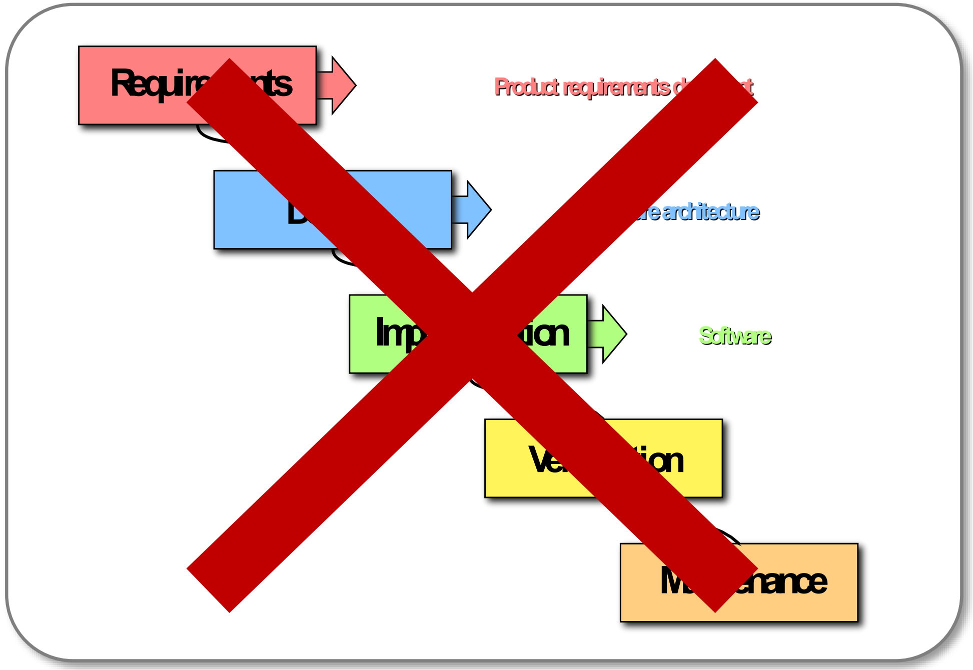
AGILE SOFTWARE DEVELOPMENT

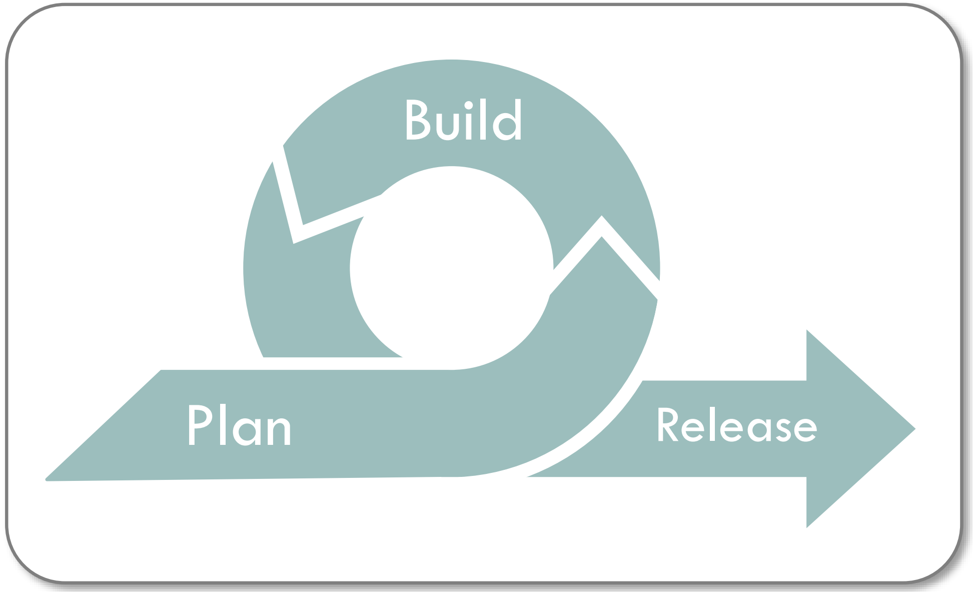
Born from frustrations with waterfall methods

- Upfront planning
- Well known and fixed requirements
- Sequential phases

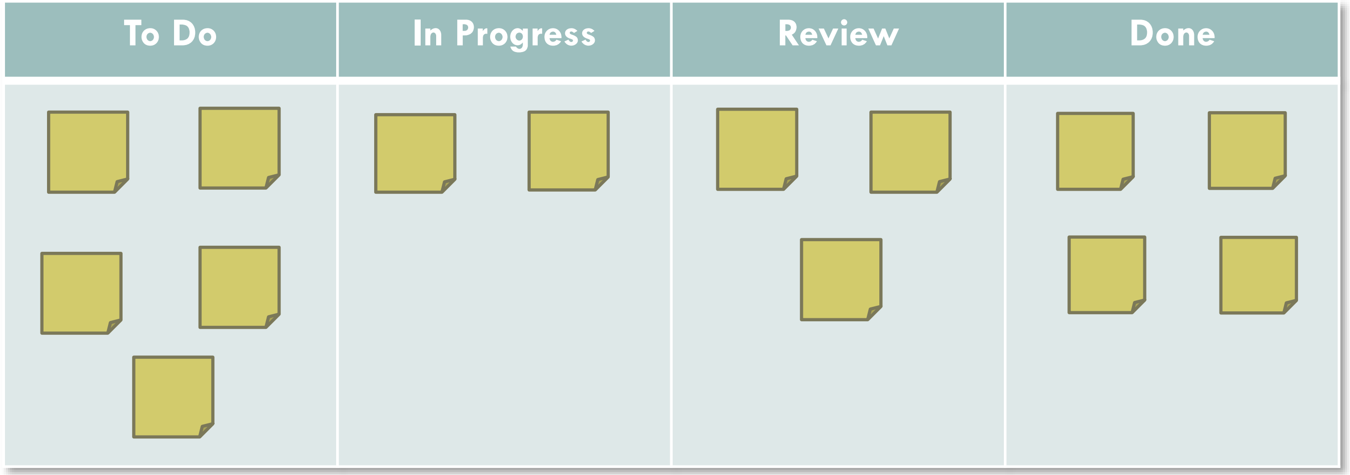
Agile methods

- Iterative, incremental delivery with feedback
- Transparency, comms & self organisation
- To explore, learn and adjust expectations
- Focus on the customer and working software





AGILE APPROACHES



To Do

In Progress

Review

Done

Two approaches to prioritisation and controlling the flow of work

- **Scrum** - Time boxing a set of priority tasks into 2-3 week “Sprints”
- **Kanban** - Restricting the amount of work in progress

Work towards a Minimal Viable Product (MVP), then extend it.

- Vertical slicing to get small parts of each component working together

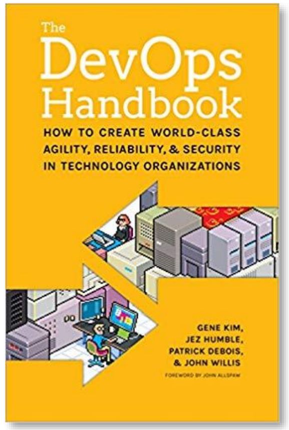
Agile Data Science With R: https://edwinth.github.io/ADSwR/

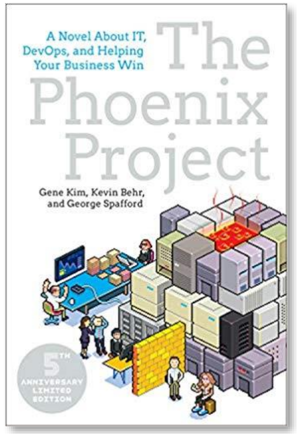
AGILE FOR DATA SCIENCE?

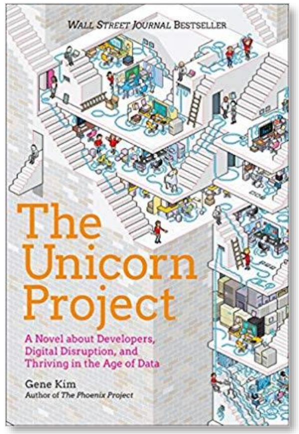
- Think about deployment from the start.
- Can then deliver quickly and repeatedly.
- Keep the first approach simple, make it end to end, then refine it
- Incremental feedback loop == The scientific method
- Allow time to test as you develop to maintain quality

- Estimating is difficult for DS projects!
- Lots of added/unforeseen tasks.

DEVOPS

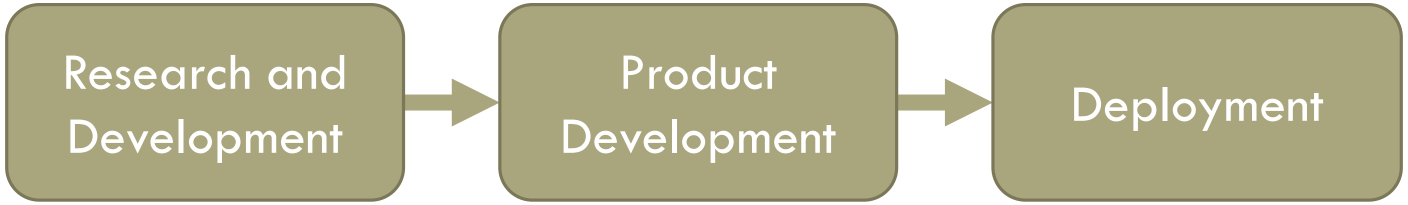



A Novel About IT, DevOps, and Helping Your Business Win
The
Phoenix
Project
Gene Kim, Kevin Behr, and George Spafford
5TH ANNIVERSARY LIMITED EDITION


WALL STREET JOURNAL BESTSELLER
The
Unicorn
Project
A Novel about Developers, Digital Disruption, and Thriving in the Age of Data
Gene Kim
Author of The Phoenix Project

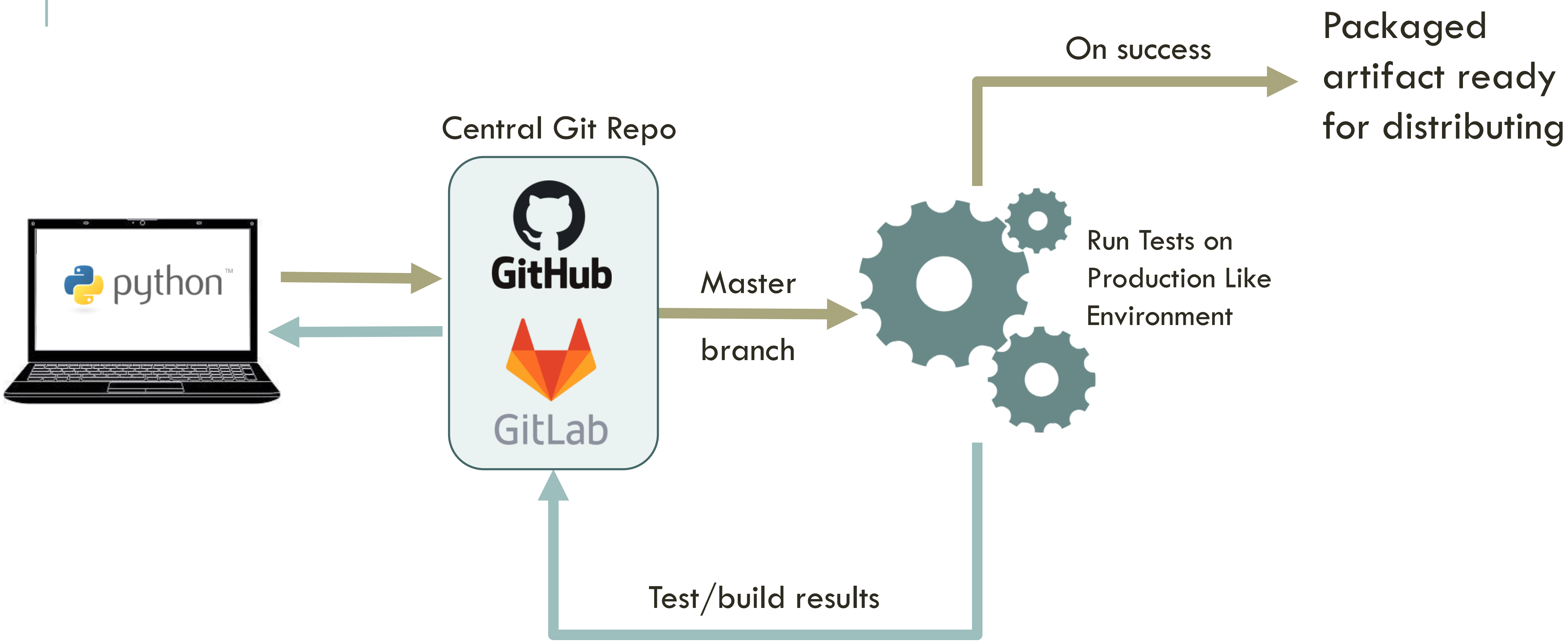
- Lean manufacturing principals applied to the IT value stream, originally between Development and Operations.
- Strives to accelerate flow and reliability throughout the value stream.
- Can be viewed as a natural extension to the Agile movement.
- Promotes ownership, collaboration, automation, self service and continuous improvement to reduce lead time, and enhance productivity.

THE 3 WAYS



1. **Flow.** Focusing on fast left to right flow of work though the value stream
   - Make work visible, reduce batch sizes, build in quality
2. **Feedback.** Enabling fast and constant flow of feedback from right to left at all stages in the value stream.
   - Problems are found and fixed quickly at the source & knowledge is captured
3. **Continual Learning and Experimentation.** Global optimisation and a scientific approach to risk taking.
   - Constant refinement, encouraging risk taking, out experiment the competition

CONTINUOUS INTEGRATION



Central Git Repo

Master branch

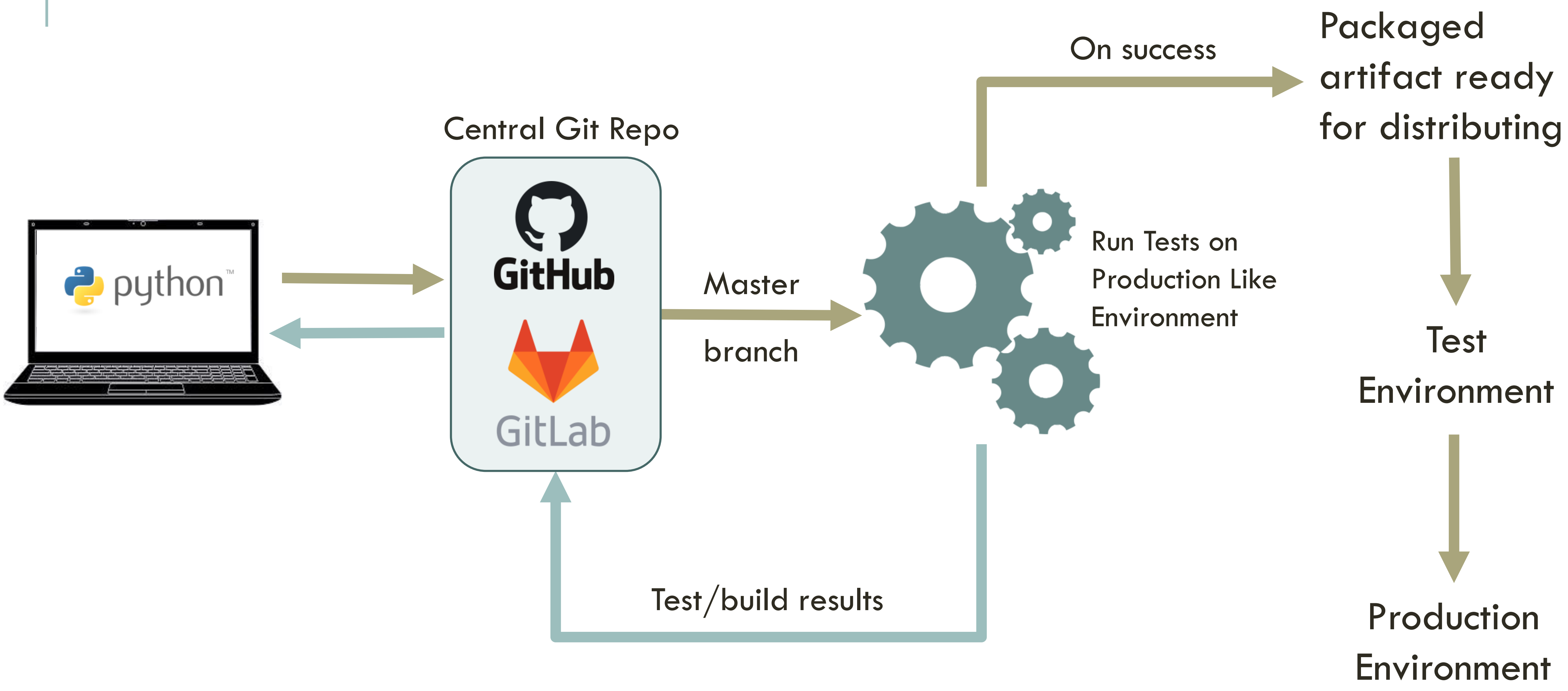
Run Tests on Production Like Environment

On success

Packaged artifact ready for distributing

Test/build results

CONTINUOUS DEPLOYMENT



JOINING THE DOTS

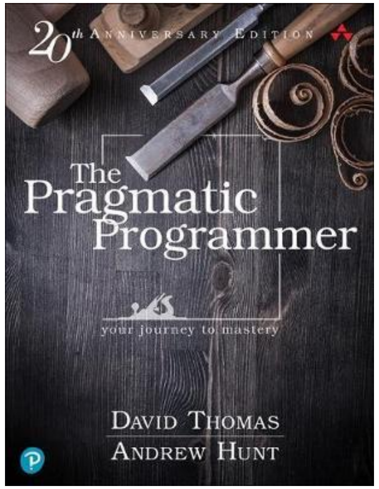
Programming for robustness and maintainability is difficult.

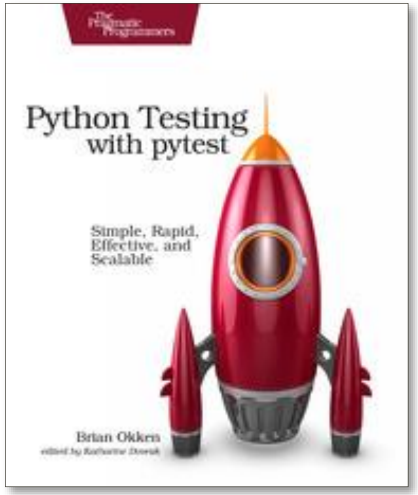
Can leverage Automated Testing, Agile and DevOps practices/methods

Enables fast and more reliable delivery of Data Science outputs

More time to do Data Science … eventually!

RESOURCES




The Pragmatic Programmers
Python Testing with pytest
Simple, Rapid, Effective, and Scalable
Brian Okken

The Agile Manifesto https://agilemanifesto.org

Agile Data Science With R: https://edwinth.github.io/ADSwR/

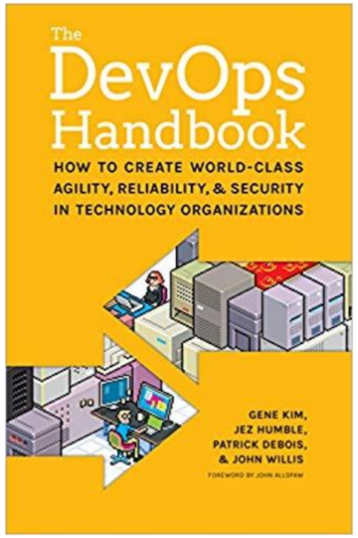
The Pragmatic Programmer

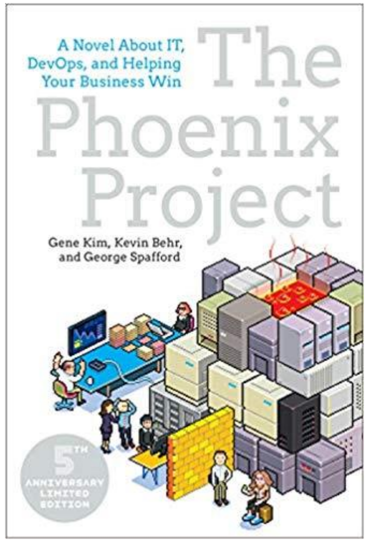
Testing with PyTest

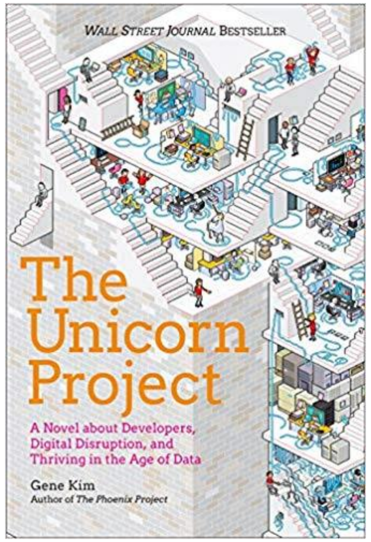
The DevOps Handbook

The Phoenix Project

The Unicorn Project


The DevOps Handbook
HOW TO CREATE WORLD-CLASS AGILITY, RELIABILITY, & SECURITY IN TECHNOLOGY ORGANIZATIONS
GENE KIM, JEZ HUMBLE, PATRICK DEBOIS, & JOHN WILLIS


A Novel About IT, DevOps, and Helping Your Business Win
The Phoenix Project
Gene Kim, Kevin Behr, and George Spafford
5TH ANNIVERSARY LIMITED EDITION


WALL STREET JOURNAL BESTSELLER
The Unicorn Project
A Novel about Developers, Digital Disruption, and Thriving in the Age of Data
Gene Kim
Author of The Phoenix Project

Chris.Musselle@malvernpanalytical.com