

Software Development in 100 Years Time

What Are the Durable Ideas?

Dave Farley

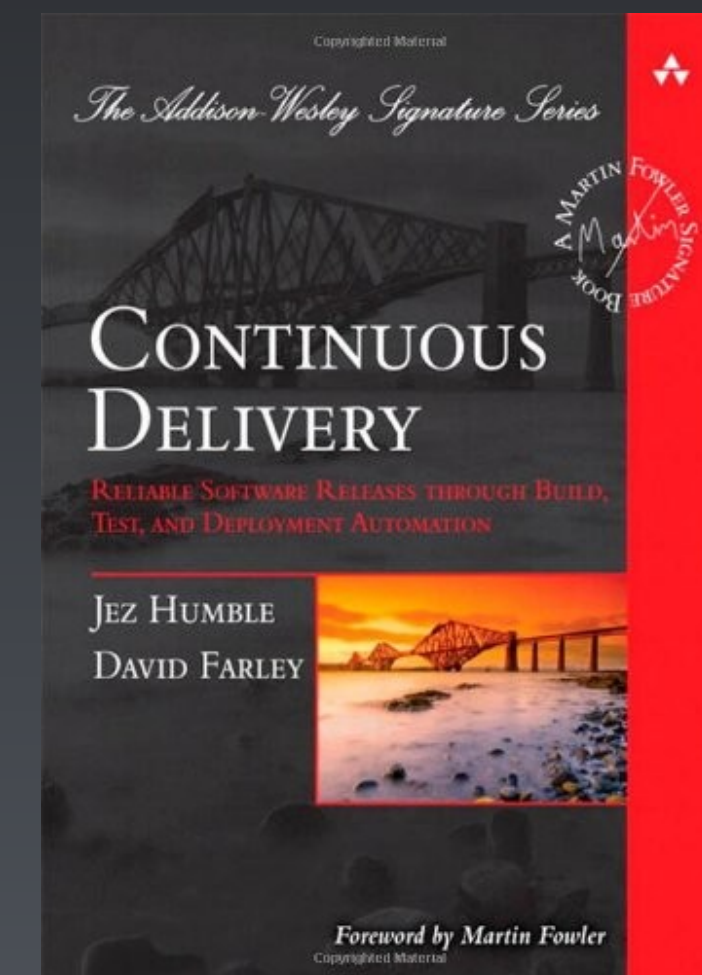
<http://www.davefarley.net>

@davefarley77

YouTube: <https://bit.ly/CDonYT>



<http://www.continuous-delivery.co.uk>



Software Development 100 Years Ago?

The image shows a wide, flat, light-colored landscape, possibly a dry lake bed or a vast field, under a clear blue sky. The horizon is straight and divides the image roughly in half. The ground is covered in small, dark specks, likely rocks or debris. The word "Nuthin'" is written in large, bold, blue letters across the middle of the image, slightly overlapping the horizon line.

Nuthin'



Nuthin'

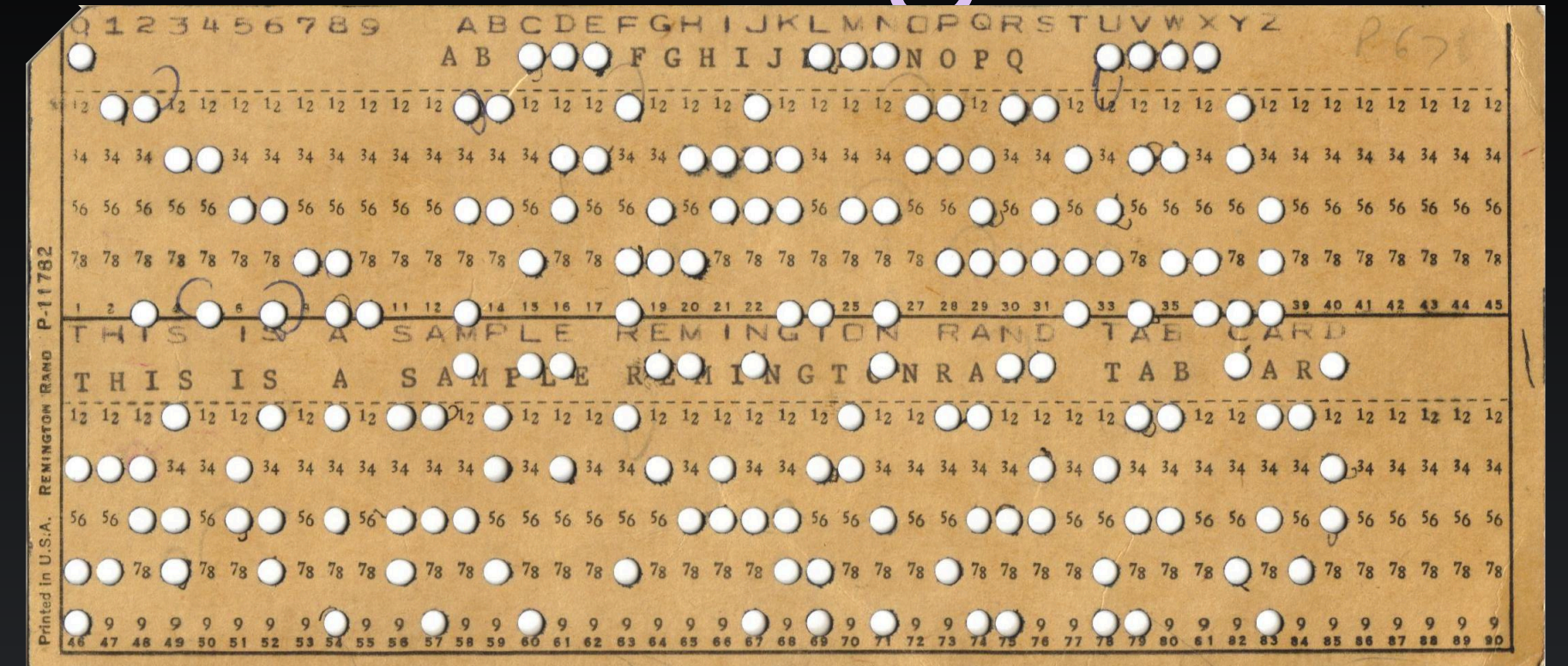
Software Development 100 Years Ago?



Software Development 100 Years Ago?



Software Development 100 Years Ago?



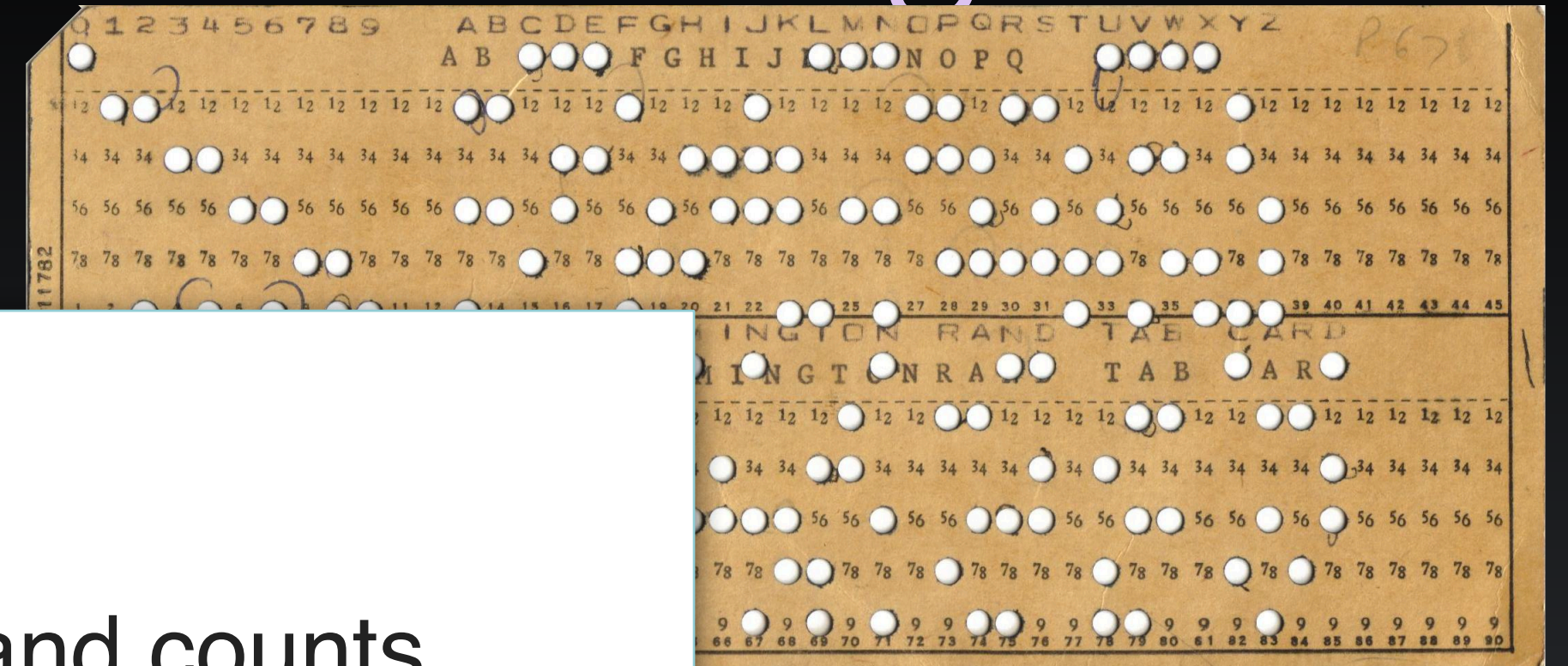
Software Development 100 Years Ago?



SELECT (filter columns)
then WHERE (filter cards, or "rows")
then maybe a GROUP BY for totals and counts,
then a SORT BY;

and then perhaps feed those back to another
set of SELECT and WHERE cycles again if needed.

Source: https://en.wikipedia.org/wiki/Tabulating_machine



Big Assumption...

Big Assumption...



Big Assumption...



**They aren't doing all
the programming!**

Lets Assume...

Lets Assume...

**Software Still
Needs to fit inside a
Human Head**

Lets Assume...

**Software Still
Needs to fit inside a
Human Head**

Lets Assume...

**Software Still
Needs to fit inside a
Human Head**

Some Ideas Are More Durable than others

Some Ideas Are More Durable than others

Language

Some Ideas Are More Durable than others

Language

Why Not?

Why Not?

**People like writing new
Languages**

Why Not?

Why Not?



Why Not?

Why Not?

**These Days Language Design is
Mostly about Fashion!**



Prediction:

There will be no great leaps in Productivity because of Programming Language

Prediction:

There won't be a huge change in the level of abstraction



Caveat:

Unless we expand the capacity of our heads!
(and we probably will!)



Some Ideas Are More Durable than others

Some Ideas Are More Durable than others

Frameworks

Some Ideas Are More Durable than others

Frameworks

Why Not?

Why Not?

**People like writing new
Frameworks too!**



Why Not?

Why Not?

**Most Frameworks Focus on the
wrong stuff!**

Why Not?

Why Not?

**Less Typing is MUCH less
important than Less Thinking!**

Prediction:



There May be big steps facilitated by “Frameworks”, but only in combination with a bigger re-think on design, architecture and programming model.

Frameworks alone are not enough!

Some Ideas Are More Durable than others

Some Ideas Are More Durable than others

**Programming
Paradigm**

Some Ideas Are More Durable than others

**Programming
Paradigm**

Some Ideas Are More Durable than others

Why Not?

Why Not?

**There seems to be something
durable about the level of detail
that we need to specify to write
code**

(Risky) Prediction:

Programming will still be Procedural, OO and Functional in parts



Controversial Opinion:

- Current fashion is to down-play the value and impact of OO
This is just a religious war
- OO was an enabling step to creating bigger, more complex systems
- OO provides an organisational approach that allows us to navigate complex problem spaces, other approaches don't do this





Prediction:

New programming paradigms will be added to better support “Services”

We are currently missing an important level of abstraction!

Some Ideas Are More Durable than others

Some Ideas Are More Durable than others

**Test Driven
Development**

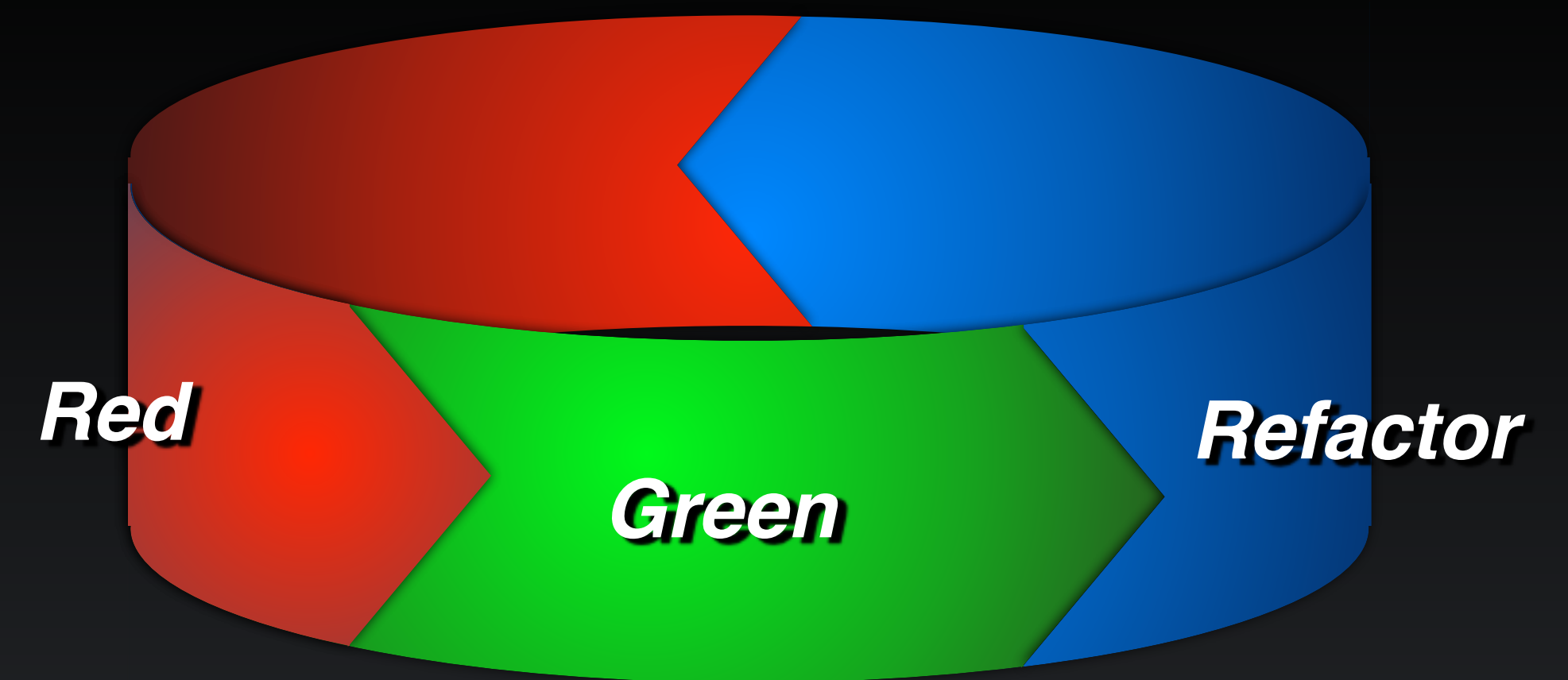
Some Ideas Are More Durable than others

**Test Driven
Development**

Why Not?

Why Not?

We got the Words wrong!



Prediction:

Test Driven Development will be a cornerstone of
“Software Engineering” but it won’t be called TDD!

(Risky) Prediction:



When a kindergarten child learns to write her first line of code, it will be done in the context of TDD (I hope)

Some Ideas Are More Durable than others

Some Ideas Are More Durable than others

Iteration

Some Ideas Are More Durable than others

Iteration
Feedback

Some Ideas Are More Durable than others

Iteration
Feedback
Incremental

Some Ideas Are More Durable than others

Iteration
Feedback
Incremental
Experimental

Some Ideas Are More Durable than others

Iteration
Feedback
Incremental
Experimental
Empirical

Some Ideas Are More Durable than others

Iteration
Feedback
Incremental
Experimental
Empirical

Why?

Why?

**These are foundational concepts.
They are grounded in our need to
learn and deepen our
understanding.**

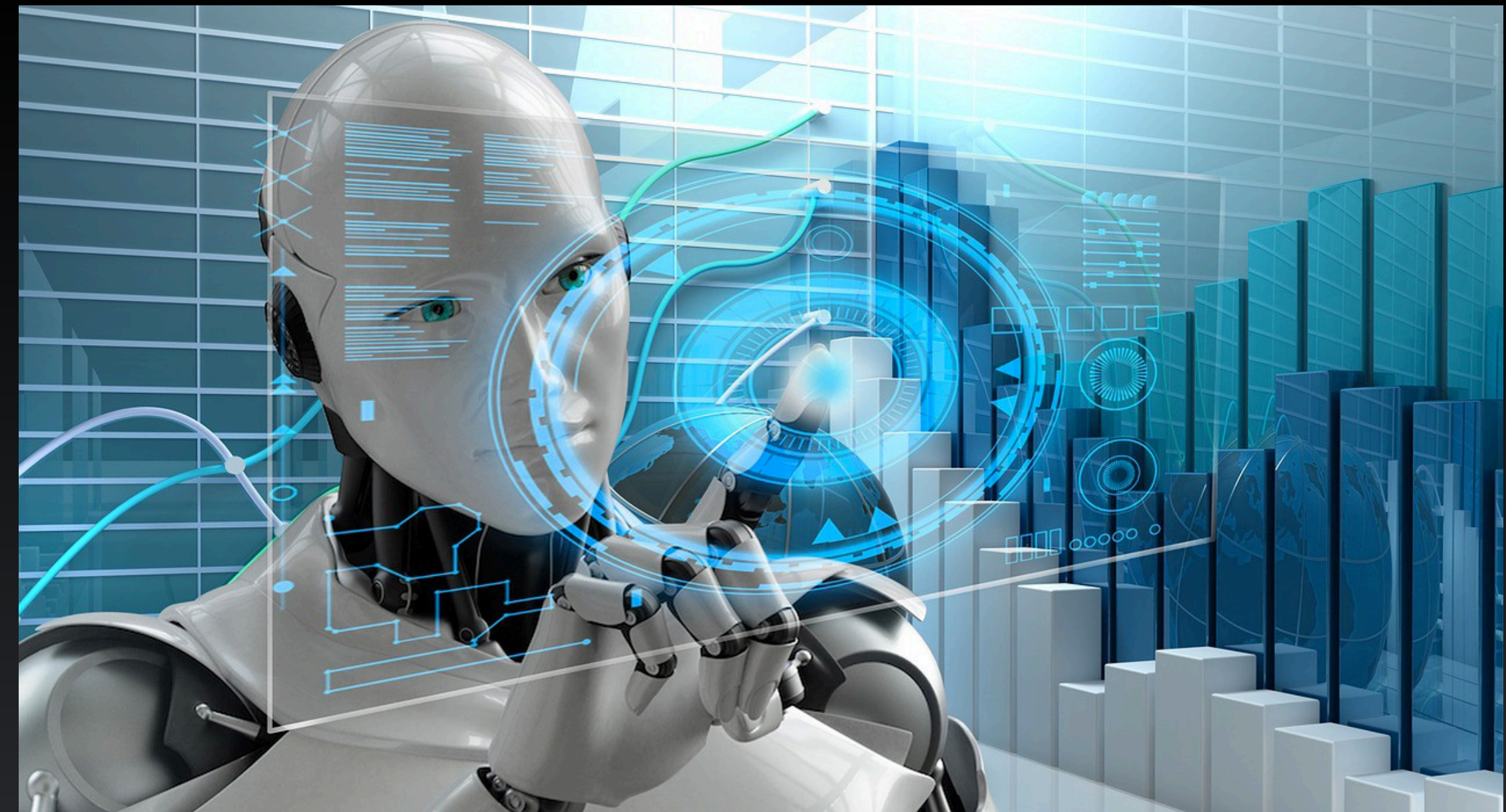
Why?

Why?

**Software Development is, and
always will be, an exercise in
learning and discovery.
We need to optimise for that!**

Prediction:

Even if the machines are writing the code, these things will still be true.



Prediction:

Software Development in 100 years time will be based on these ideas.

These are important, durable ideas and there are ***BIG STEPS*** forward in quality and efficiency to be had here.





Observation:

In 1986 Fred Brookes wrote “There is no Silver Bullet”

...but there are “Mud Bullets!”, some things just don’t work!

We need to eliminate those things.

Some Ideas Are More Durable than others

Some Ideas Are More Durable than others

Modularity

Some Ideas Are More Durable than others

Modularity

Separation of Concerns

Some Ideas Are More Durable than others

Modularity
Info Hiding
Separation of Concerns

Some Ideas Are More Durable than others

Modularity
Info Hiding
Separation of Concerns
Loose-Coupling

Some Ideas Are More Durable than others

Modularity
Info Hiding
Separation of Concerns
Loose-Coupling
Cohesion

Some Ideas Are More Durable than others

Modularity
Info Hiding
Separation of Concerns
Loose-Coupling
Cohesion

Why?

Why?

We need to decompose ideas into pieces that are small enough to fit into a Human Head

Why?

Why?

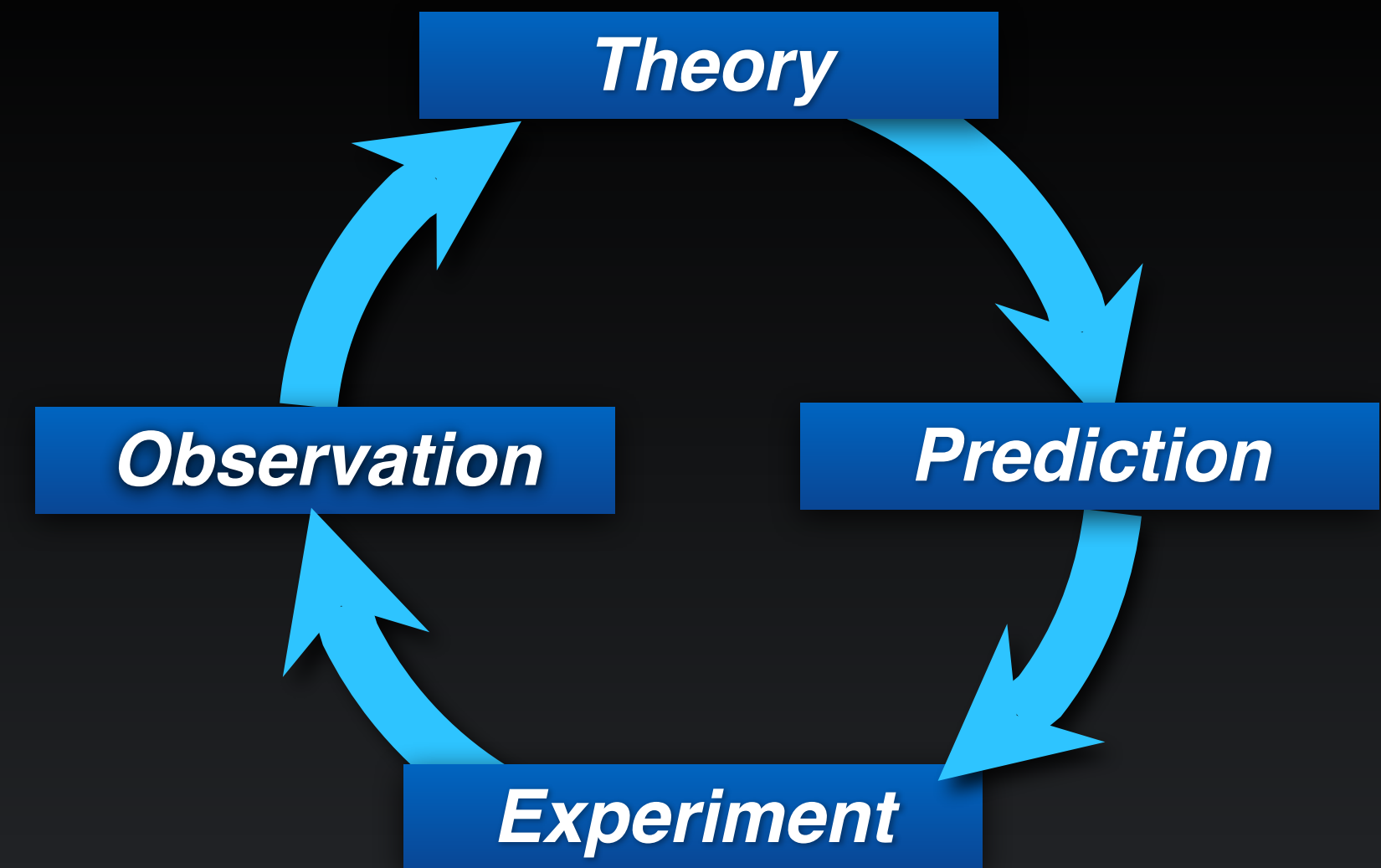
**SW Dev is about learning and
managing complexity**

Prediction:

Even if we mechanically extend our cognitive function, we will still need to partition problems to fit our (new expanded) heads



Observation:



Software Development is about Learning and Discovery.

Humanity's best approach to learning and discovery is the ***Scientific Method***

Hence:

Iteration, Feedback, Incrementalism, Experimentation and Empiricism!

Observation:



Software Development is also about managing complexity.

Humanity's best approach to complexity is compartmentalising things and reducing coupling.

Hence:

Modularity, Information Hiding, Separation of Concerns,
Loose-Coupling, Cohesion

Some Ideas Are More Durable than others

Some Ideas Are More Durable than others

Continuous Delivery

Some Ideas Are More Durable than others

Continuous Delivery

Why?

Why?

**CD is based on the application of
the Scientific Method to SW Dev.**

Why?

Why?

**The Scientific Method is
Humanity's Best Problem Solving
Approach!**

It isn't going out of fashion!

Why?

Why?

**CD has at it's heart:
Iteration, Feedback, Incremental
Design, Experimentation and
Empiricism**

Why?

Why?

**CD, Through Automated Testing,
Encourages:
Modularity, Info Hiding,
Separation of Concerns,
Abstraction and Cohesion**

Prediction:



Even if the machines do write the code, they will adopt CD

Q&A



<http://www.continuous-delivery.co.uk>

Dave Farley

<http://www.davefarley.net>

@davefarley77

YouTube: <https://bit.ly/CDonYT>

