

# Call It Science!

---

LAURA MOORE – ATLANTA SCRUM USER GROUP – AUG 2018

# Call it Science!

➤ PLEASE TAKE 1 SUIT OF CARDS (A-K) FROM A DECK

---

LAURA MOORE – ATLANTA SCRUM USER GROUP – AUG 2018

“Most developers think that the primary benefit derived from TDD is the comprehensive set of unit tests you end up with.”<sup>1</sup>

“every attendee perceived TDD as being about correctness: ‘code free of defects’, ‘automated regression testing vs. manual’, ‘changing code without fear of breaking it,’ etc.”<sup>2</sup>

**“So as long as you write tests - before or after the production code doesn't really matter IMHO”**



**Tim Ottinger**

@tottinge

Following



I think people misunderstand MVPs and Evolutionary Design as "stepwise building of a solution" not "progressively learning market needs"

11:31 AM - 28 Apr 2016

And Technology

we naturally gravitate to the comfort of being right, especially when that's what we're being paid for...

If the result confirms the hypothesis, then you've made a *measurement*. If the result is contrary to the hypothesis, then you've made a *discovery*.

*-Enrico Fermi*

# Test Driven is more than just *Testing*?

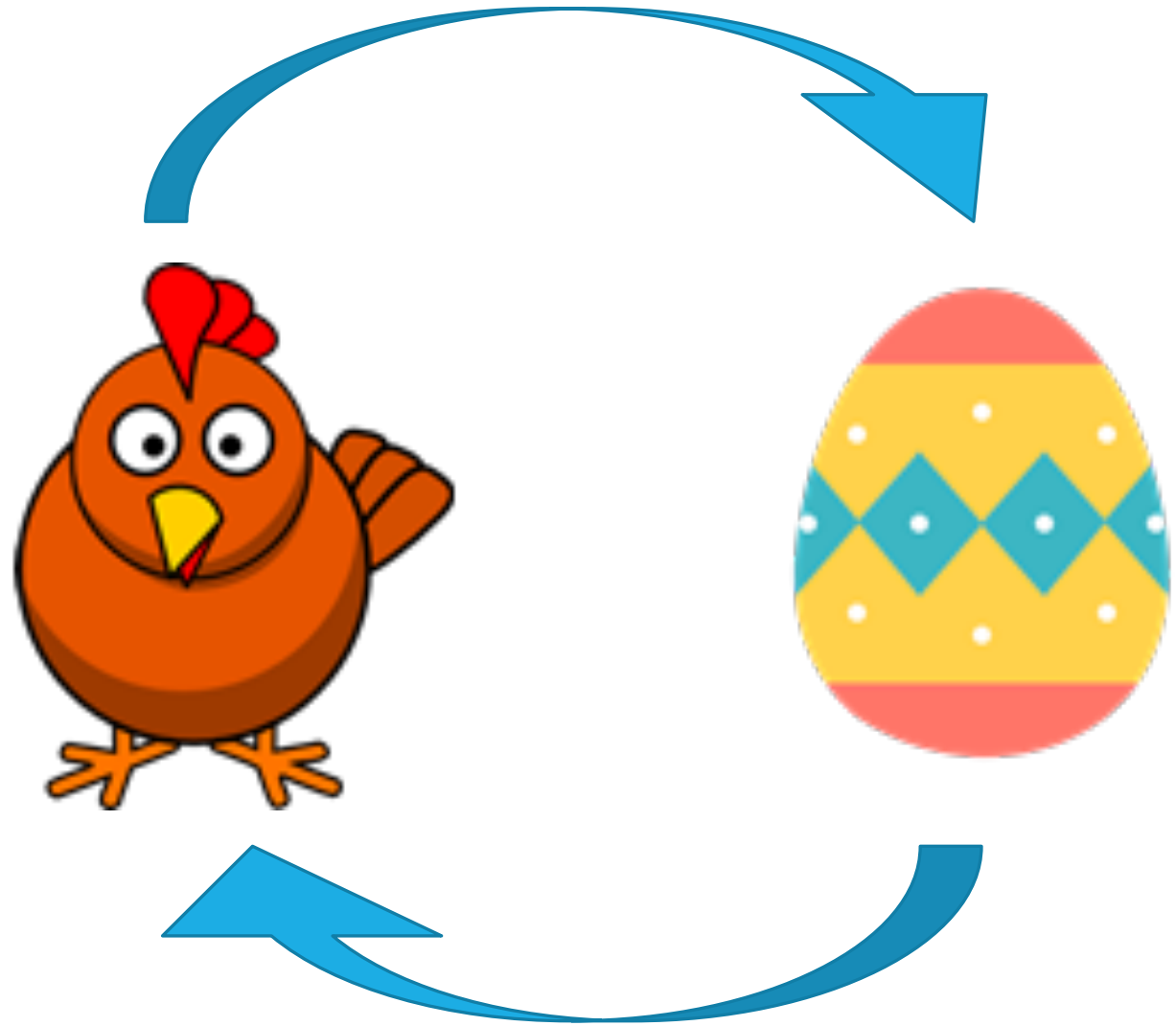
---

#WHY\_THE\_CARDS?

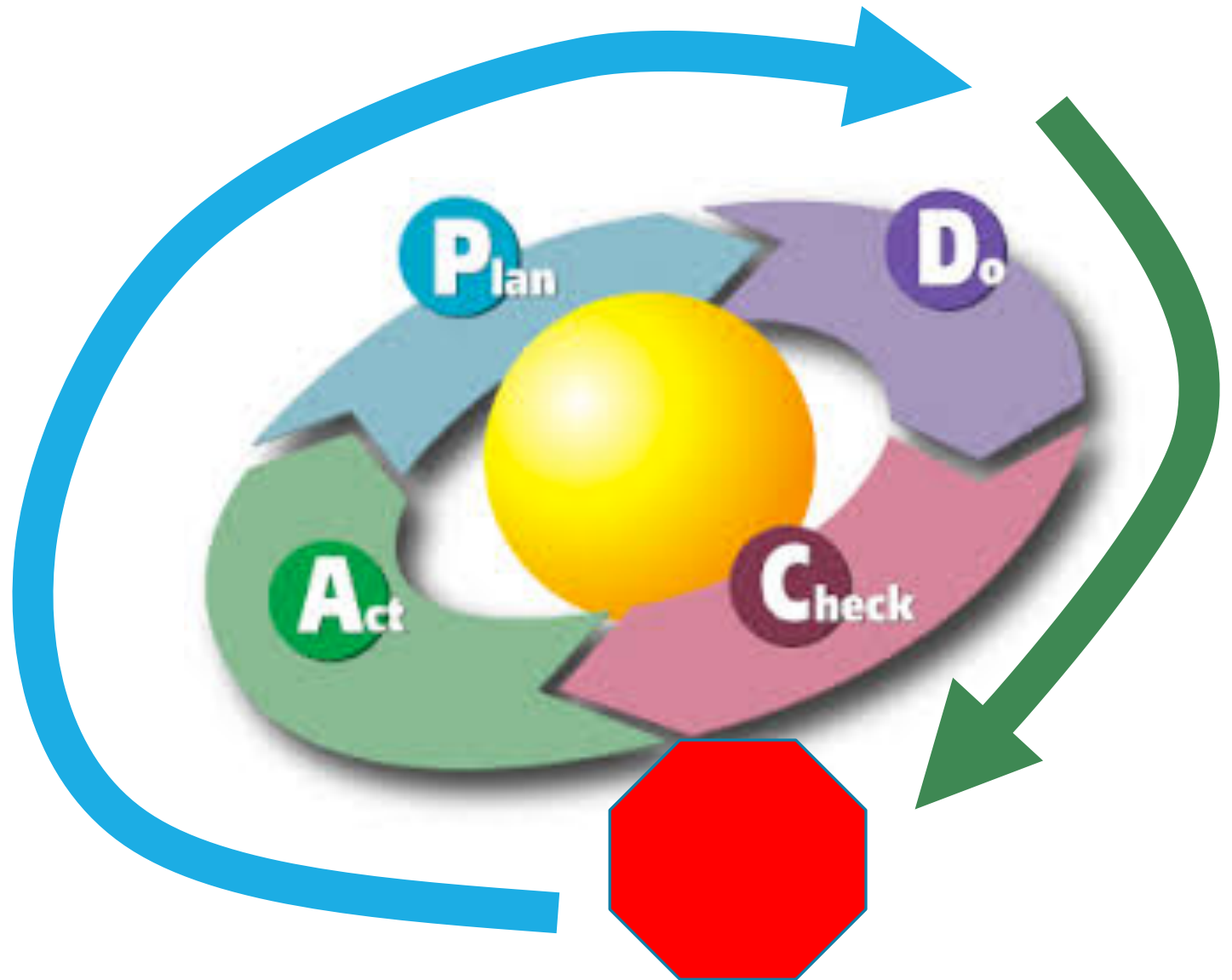
# Which Evolves First?

“We’re in the business of shipping eggs, why does my backlog keep filling up with stories about chickens?”

- A. Frustrated P.O.



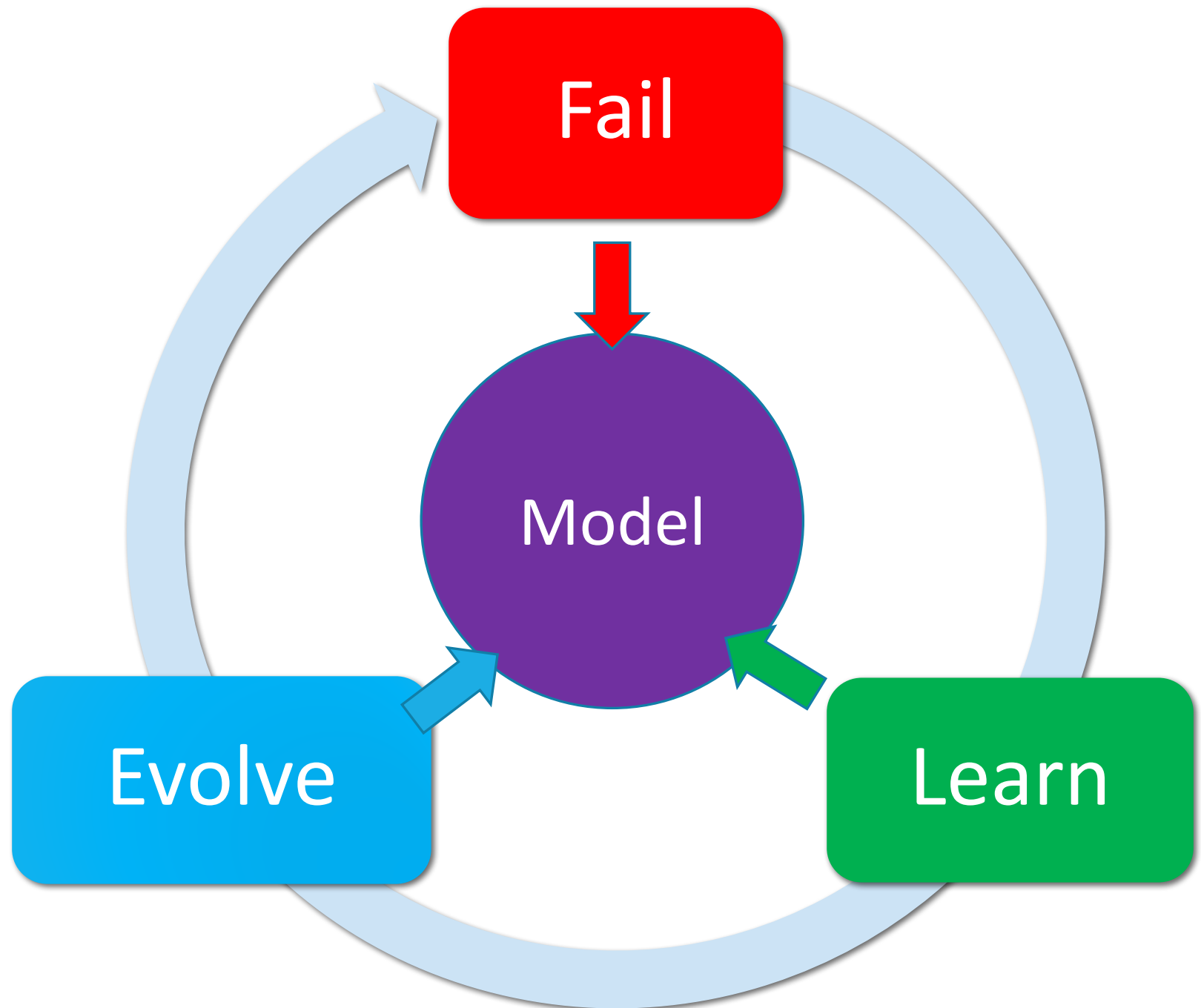
A common iteration plan....





# Mental Model

“If you want to do a better job of paying attention to what really matters, narrate your life as it’s occurring.” (3)



# How do you *Test* for an unknown?

---

#WHY\_THE\_CARDS?

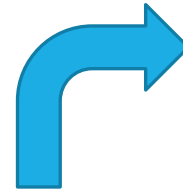
All you have to do...

Arrange the cards so that running them through the following

1. All Face Down
2. 1<sup>st</sup>, 3<sup>rd</sup> ... Card face up in front of you
3. 2<sup>nd</sup>, 4<sup>th</sup> ... Card back under the deck
4. Repeat for 13 cards

Results in All Up in Suit Order

Arrange suit of 13 cards in ????



1<sup>st</sup> card up = A  
3<sup>rd</sup> card up = 2  
Continue....



2<sup>nd</sup> Card Under  
4<sup>th</sup> Card Under  
Continue....



Final result when all cards face up

# Round 1

---

1

2

3

4

What's your style?

Prototype

Brute Force

Reverse Engineer

# Round 2

---

1

2

3

4

Did any of these help?

Changing Tactics

Pairing / Mobbing

Outsource

No Spoilers!

Post a pic or vime of your best attempt to order the cards and tag me. I'll send you the lab solution.



# Steel Thread

“tiny implementation of the system that performs a small end-to-end function. It need not use the final architecture, but it should link together the main architectural components. The architecture and the functionality can then evolve in parallel.” (6)

- [Alistair Cockburn](#).



Continuous Testing is  
*more* than automation.

---

#AGILE\_CRAVES\_FEEDBACK

Before there was  
code there was  
testing.....

The W. EDWARDS  
Deming  
Institute



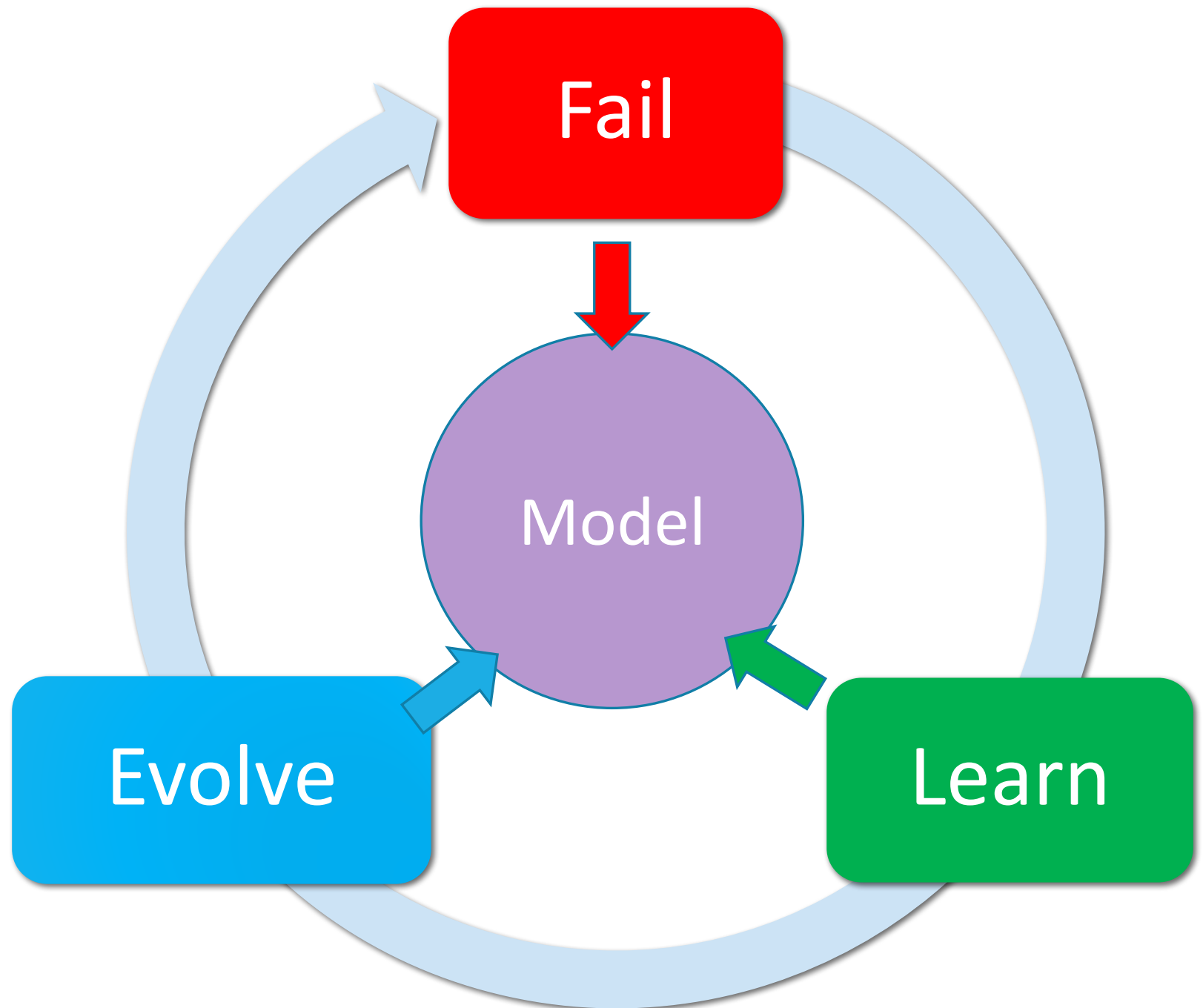
Inspection does not improve the quality, nor guarantee quality. Inspection is too late. The quality, good or bad, is already in the product. As Harold F. Dodge said, “You can not inspect quality into a product.”

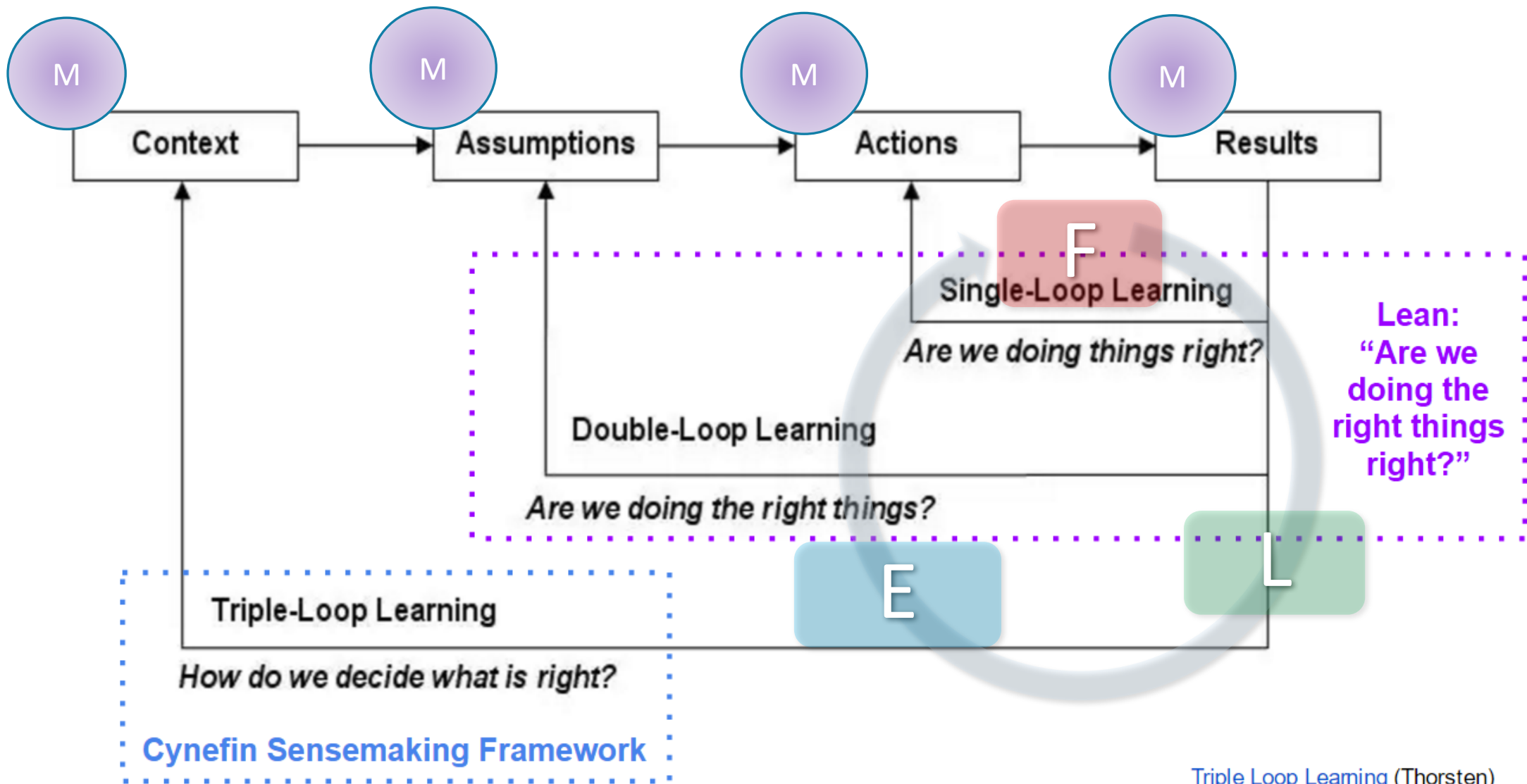
**W. Edwards Deming**

source: [quotes.deming.org/10157](https://quotes.deming.org/10157)

# Your Mental Model

“If you want to do a better job of paying attention to what really matters, narrate your life as it’s occurring.” (3)

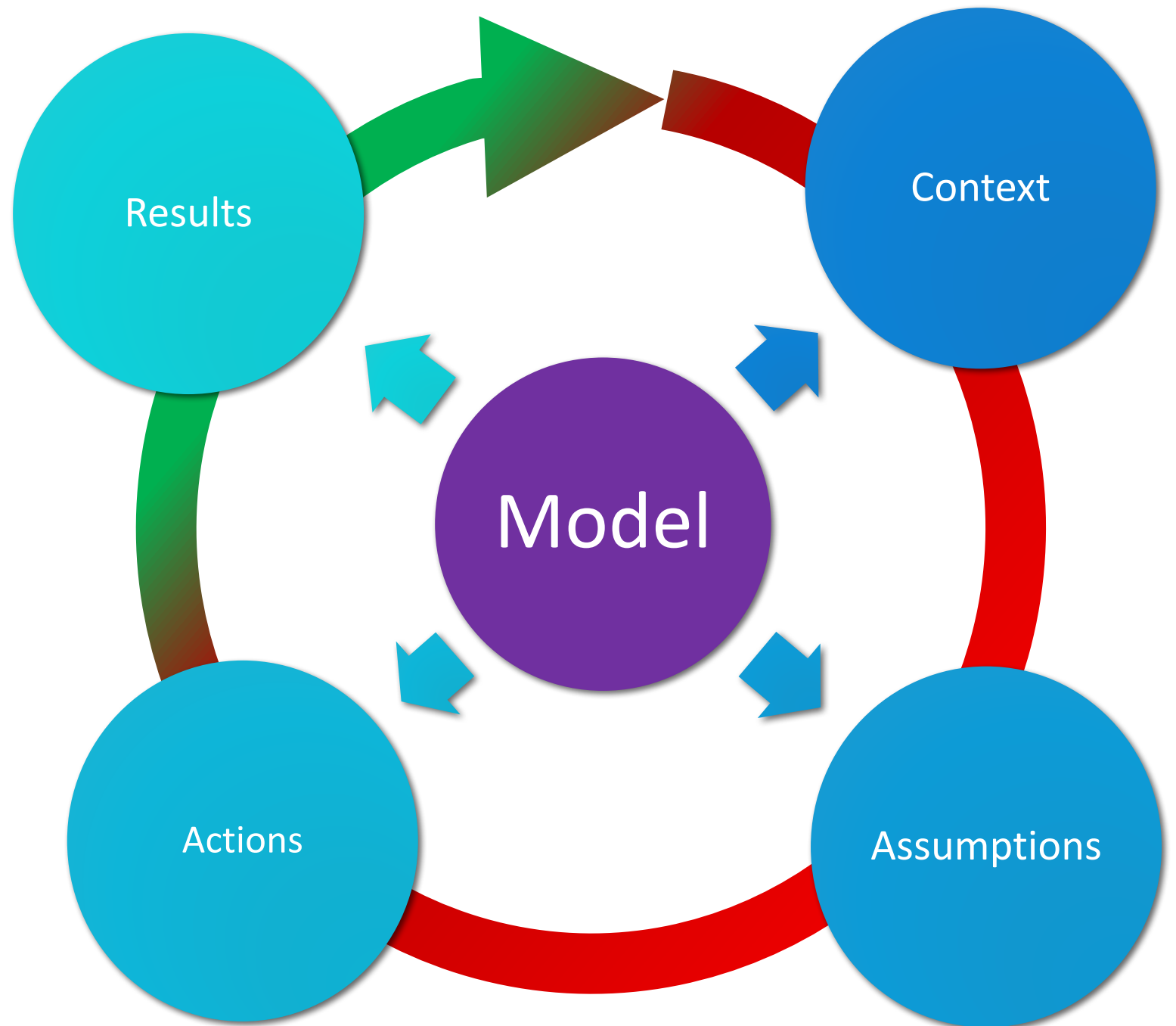




Jack Caine : **Dumbing Down The Message**

Red, Green ->  
Refactor the  
Model

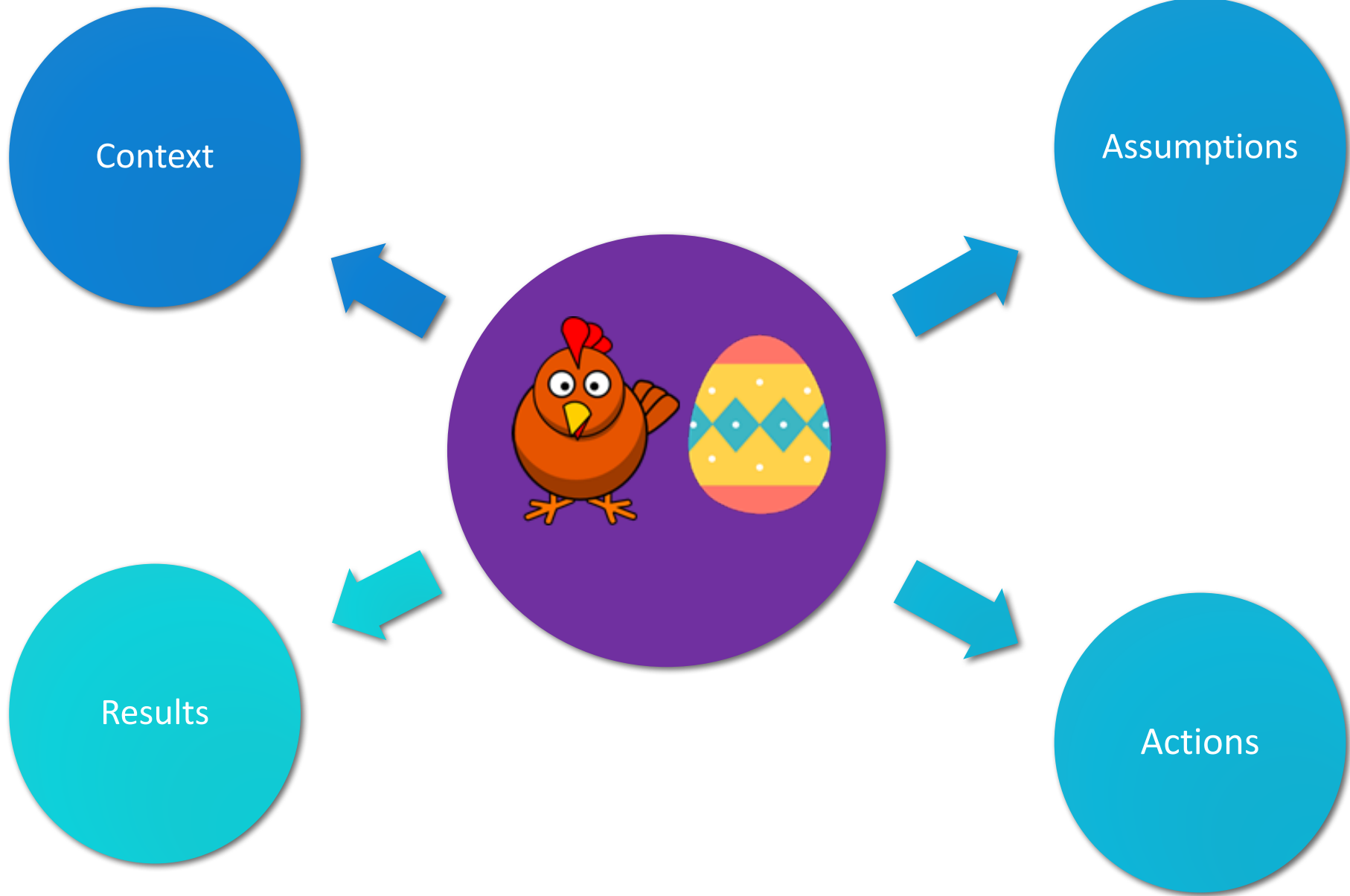
“In a world of endless  
data how do you tell  
the difference between  
garbage and good  
sense? “(3)



# Testing is a *Team* Sport

---

#EVERY\_ACTIVITY\_A\_TESTING\_ACTIVITY





“If you want to do a better job of paying attention to what really matters, narrate your life as it’s occurring.”(3)

Testing your Stories

Testing your Plans

Testing Assumptions

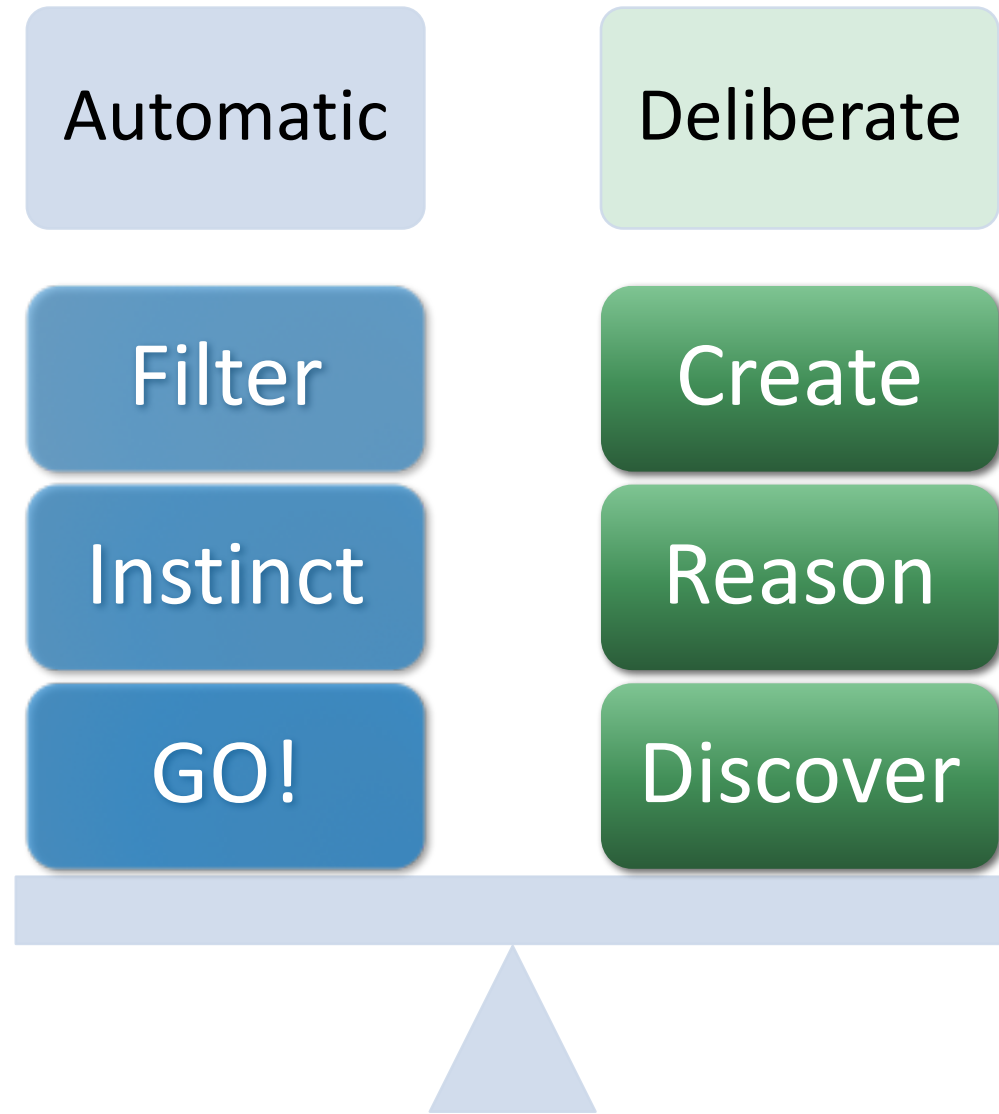
# Stories: INVEST for both brains

---

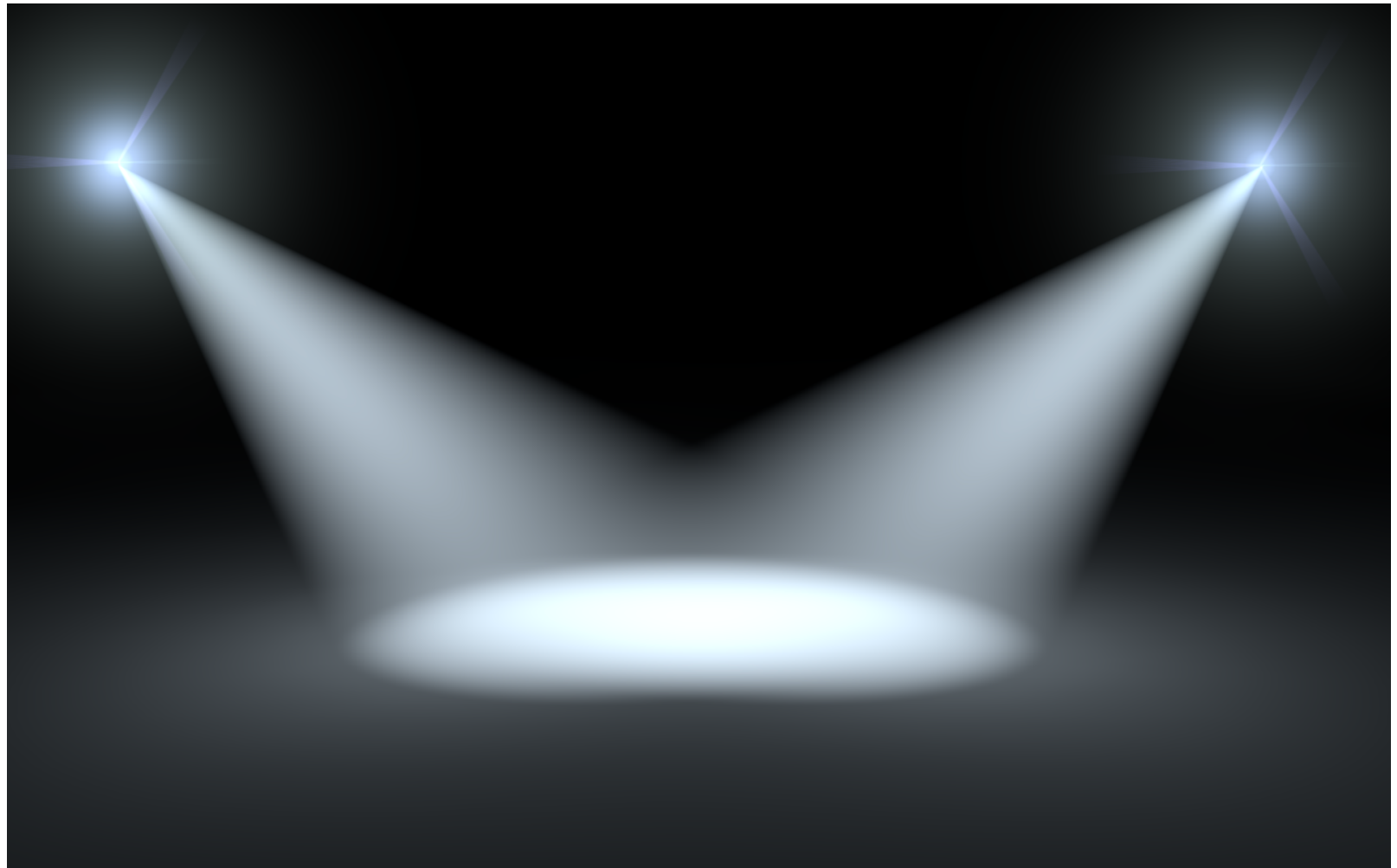
#CONTEXT IS KING

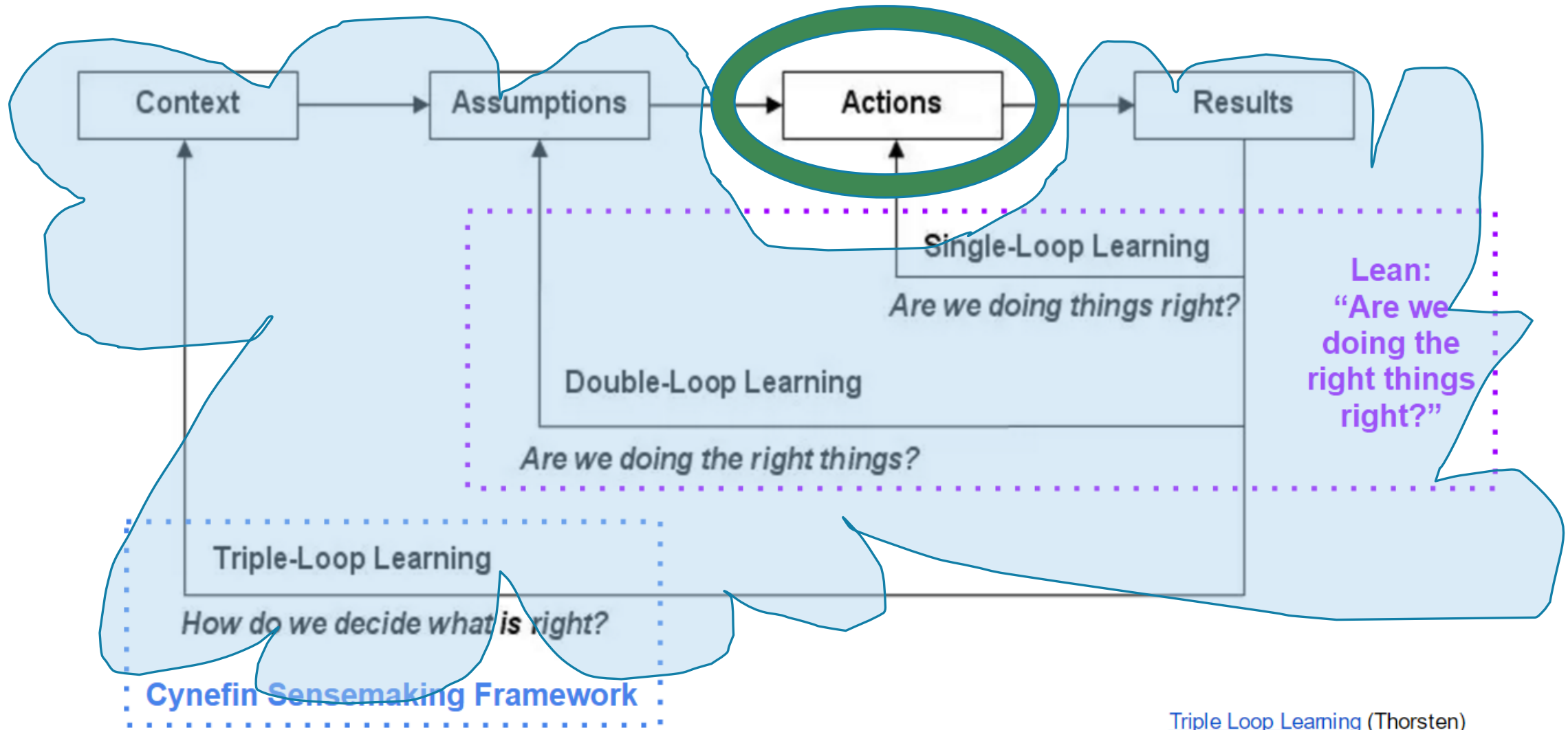
# Two Brain System

“... what it takes us to be smart and effective in handling challenges”  
(5)



“You can think about your brain's attention span like a spotlight that can go wide and diffused, or tight and focused”(4)





Jack Caine : **Dumbing Down The Message**

Right Sized  
Stories

INVEST

Wide enough to be  
motivational

Specific enough to be  
controllable

Risky enough to be  
educational

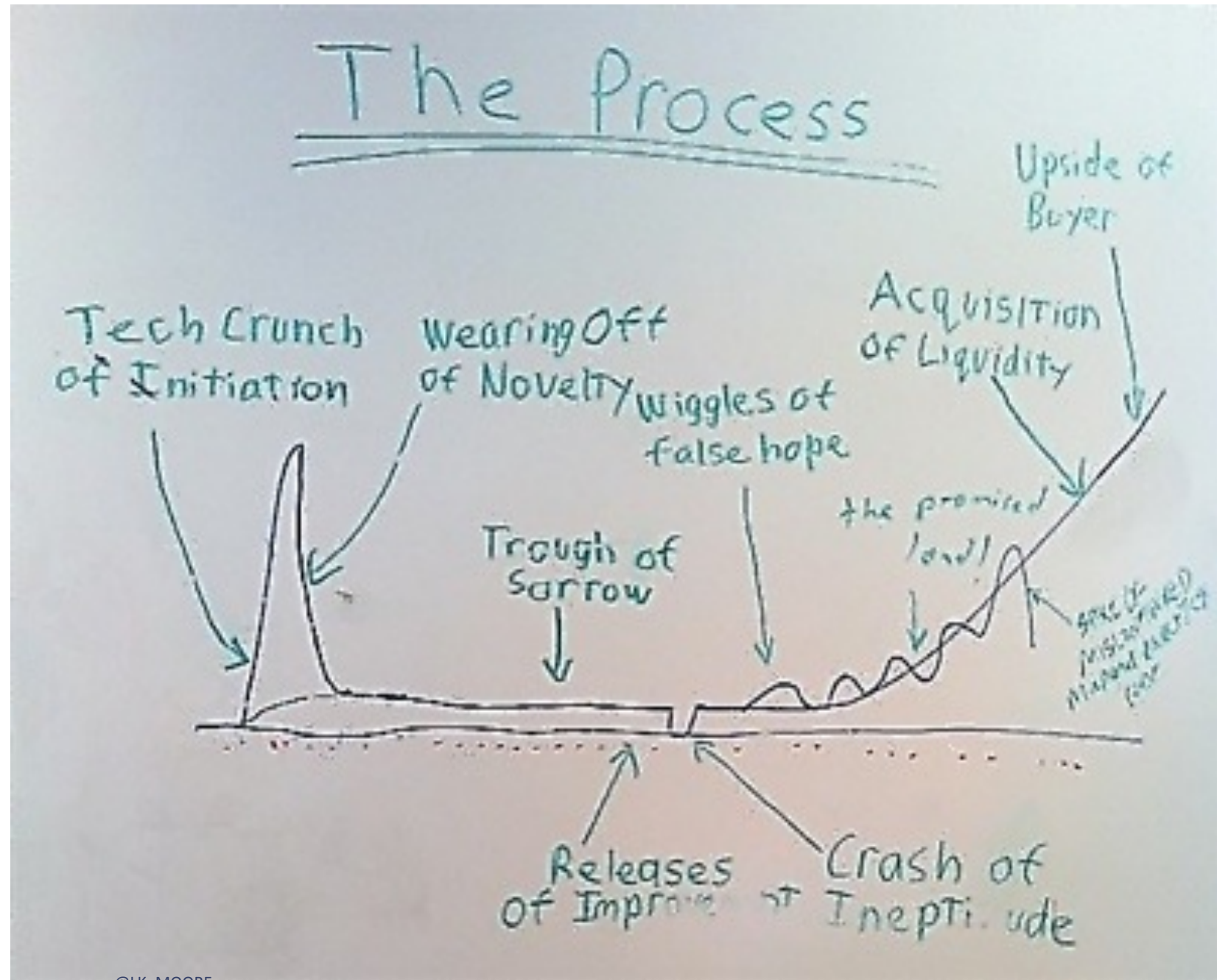
# Plans: 20/80 Vison

---

#SETTING PRIORITY OF ACTION

# Plan to Learn

A learning loop unrolled  
(5)





# Ferriss DiSSS Model (6)



# Steel Thread

“tiny implementation of the system that performs a small end-to-end function. It need not use the final architecture, but it should link together the main architectural components. The architecture and the functionality can then evolve in parallel.” (6)

- [Alistair Cockburn](#).



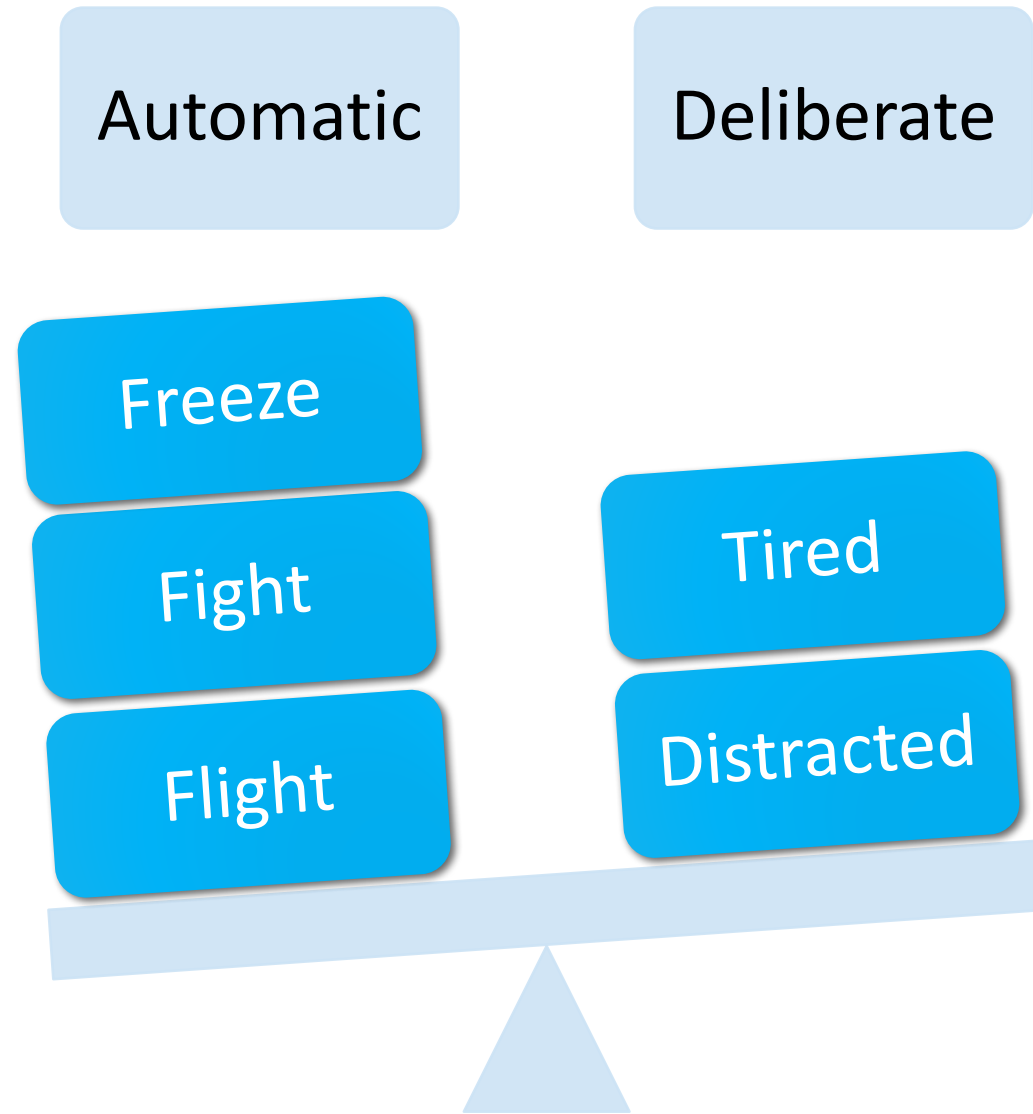
# Results: Reality Checks

---

#TRUTH SEEKING CHARTER

# Avoiding Traps

“... we are not our smartest when in automatic, defensive mode” (4)



# Cognitive Bias

“Our automatic system will quietly take control and rush us to the easiest answer available” (4)

“It’s Obvious”

“I recently heard ...”

“Makes Sense to me”

“Stick to what we know”

“Everyone Agrees”

“It’s the only option”

## Cross Check<sup>(5)</sup>

Don't Default

Engineer Dissent

Conduct a Pre-Mortem

## Cognitive Closure

*“making a decision and moving on to the next question feels productive. It feels like progress.”(3)*



## Ways to Cross Check <sup>(4)</sup>

Freshest Eyes First

Issue Tree

Change Perspective



# Notes and References

---

1. <http://blog.testdouble.com/posts/2014-01-25-the-failures-of-intro-to-tdd.html>
2. Jack Caine: Dumbing Down the Message <http://www.carmelrush.com/index.php/component/content/article?id=21:dumbing-down-the-message>
3. Charles Duhigg : Smarter Faster Better: The Transformative Power of Real Productivity
4. Caroline Webb: How to Have a Good Day
5. Timothy Ferriss: The 4-Hour Chef
6. <https://www.industriallogic.com/blog/evolution-cupcakes-and-skeletons/>

Sources not directly quoted:

- a) Code as Design Part 1 [www.developerdotstar.com](http://www.developerdotstar.com)
- b) Joe Jackson: A World on Fire: A Heretic, an Aristocrat, and the Race to Discover Oxygen
- c) Chris Britton: Designing the Requirements: Building Applications that the User Wants and Needs
- d) Steve Freeman: Growing Object-Oriented Software, Guided by Tests
- e) Richard Sheridan: Joy, Inc.: How We Built a Workplace People Love
- f) From Developer to Architect: Patterns, Architecture Types, Soft Skills, and Continuous Delivery –Neil Ford and Mark Richards
- g) Art Markham: Smart Change
- h) Neil Ford: <https://www.ibm.com/developerworks/java/library/j-eaed2/index.html>