

# SYSTEMS THINKING

---

FOR DEV TEAMS

**THANK YOU!!!**

**HELP ME!!!**



**Tweets**  
**120**

**Following**  
**118**

**Followers**  
**118**

**Likes**  
**213**

**Lists**  
**0**

**Moments**  
**0**

**Jason Blackhurst**

@alsoKnownAsSlim

# SLIM

---

HUSBAND OF 1, FATHER OF 2,  
FAIRLY TECHNICAL.

**BEHIND THE CURTAIN**

---

# **!! DISCLAIMERS DISCLAIMERS DISCLAIMERS !!**

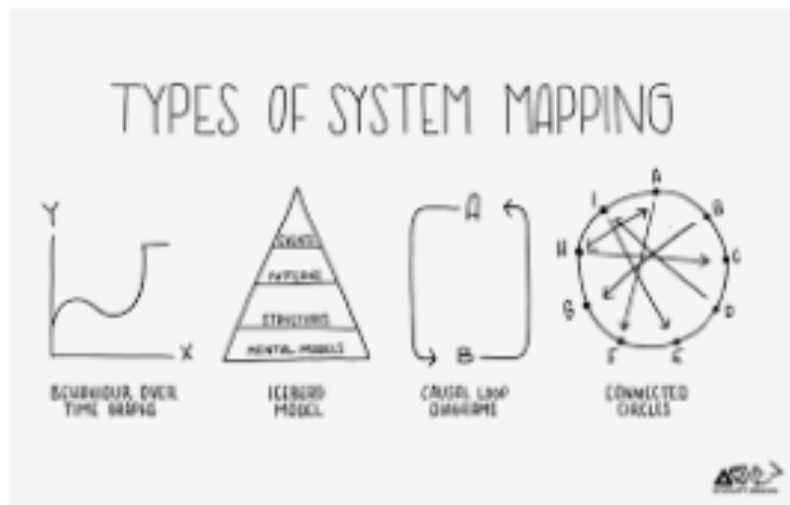
Opinions seen here are mine and do not necessarily reflect those of:

Revel IT, FastSwitch, AEP, Charge, Agile Sashimi, Agility Partners, Rob Kinyon, MCPc, Fuse, Cardinal Health, ICC, McGraw-Hill, Jez Humble, Gene Kim, ThoughtWorks, Dan North, James Shore, Martin Fowler, craigles75, STeve Shary, Richard Bradshaw, Google, Wikipedia, Amazon, AWS, any Federal Gov't branch, my mother, you, anyone else or their mother.\*

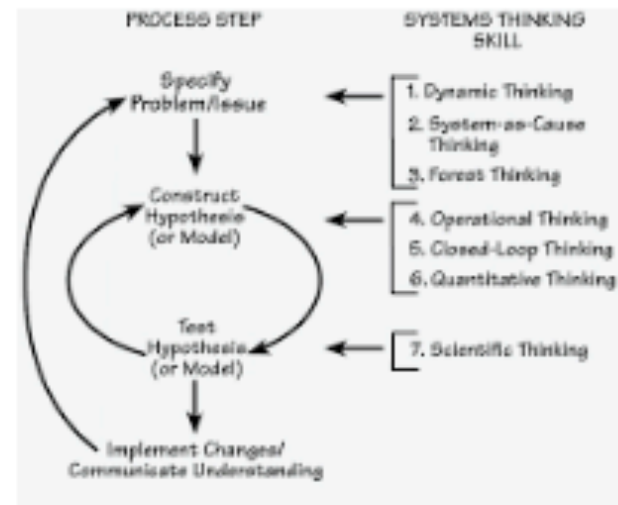
\*Opinions/views do not necessarily differ either.

**WHAT IS  
SYSTEMS  
THINKING?**

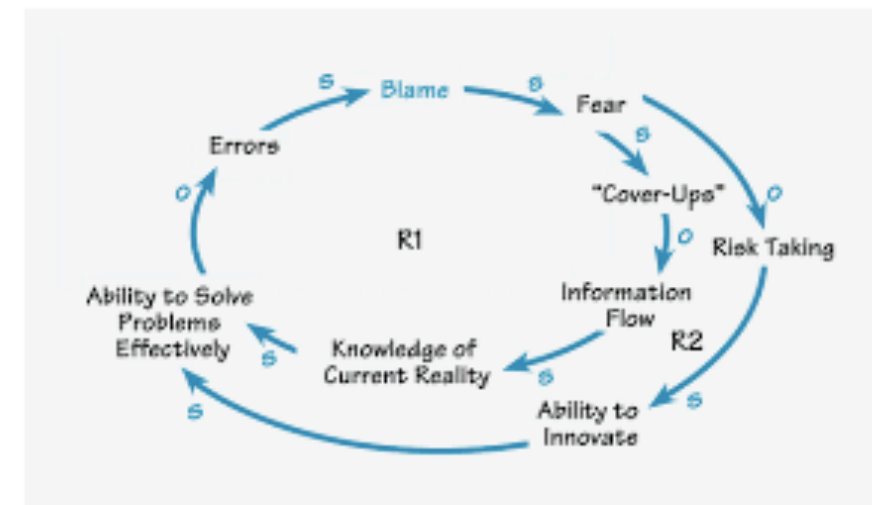




Tools for Systems Thinkers: The 6 ...  
medium.com



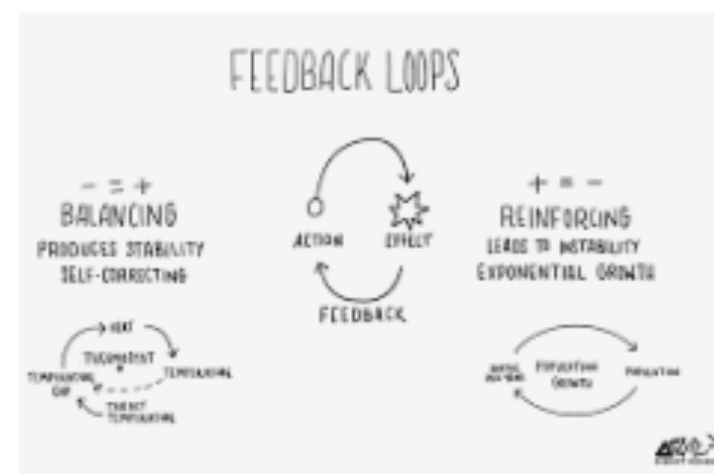
The Systems Thinker – The "Thinki...  
thesystemsthinker.com



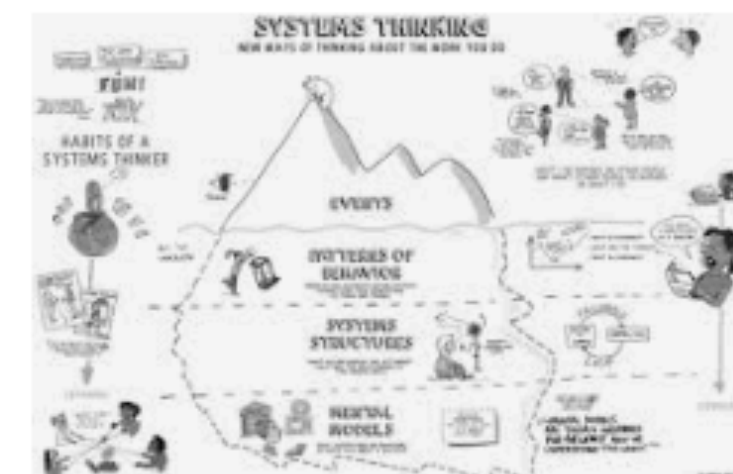
The Systems Thinker – Systems Thinking ...  
thesystemsthinker.com



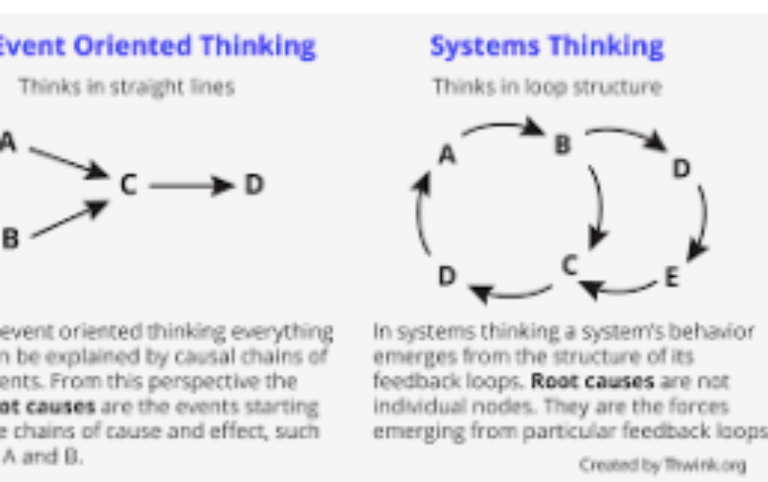
The Systems Thinker – Systems Thinking ...  
thesystemsthinker.com



Tools for Systems Thinkers: The 6 ...  
medium.com



Systems Thinking - The Helix Group ...  
kazimierzgozdz.com



Event Oriented Thinking - Tool/Concept ...  
thwink.org



Systems Thinking | Global Advocacy ...  
advocacy.thp.org



Systems Thinking: Understanding the ...  
ualberta.ca

---

# SYSTEMS THINKING

- ▶ Study of cohesive conglomerations of interrelated and interdependent parts
- ▶ Evaluate the situation as it relates to the System.
- ▶ Expand the boundaries out a level.

THE MOTHER OF FAMILY THERAPY

---

**VIRGINIA SATIR**

REVISED & EXPANDED EDITION

*The* DESIGN  
*of* EVERYDAY  
THINGS

DON  
NORMAN



**HOW ABOUT  
DEVOPS?**



### ► **Wikipedia:**

DevOps (a clipped compound of "software DEvelopment" and "information technology OPerationS") is a term used to refer to a set of practices that emphasize the collaboration and communication of both software developers and information technology (IT) professionals while automating the process of software delivery and infrastructure changes.[1][2] It aims at establishing a culture and environment, where building, testing, and releasing software can happen rapidly, frequently, and more reliably.[3][4][5]

### ► **Amazon:**

DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.

**TL;DR**

**A NEW**

**PERSPECTIVE**



**DEVOPS IS DEVELOPMENT  
WHERE OPERATIONS IS THE  
PROBLEM DOMAIN.**

**Rob Kinyon**

# AN AGILE PERSPECTIVE

**BUSINESS**

**DEVELOPMENT**

**BUSINESS**

**DEVELOPMENT**



**BUSINESS**

**OPERATIONS**

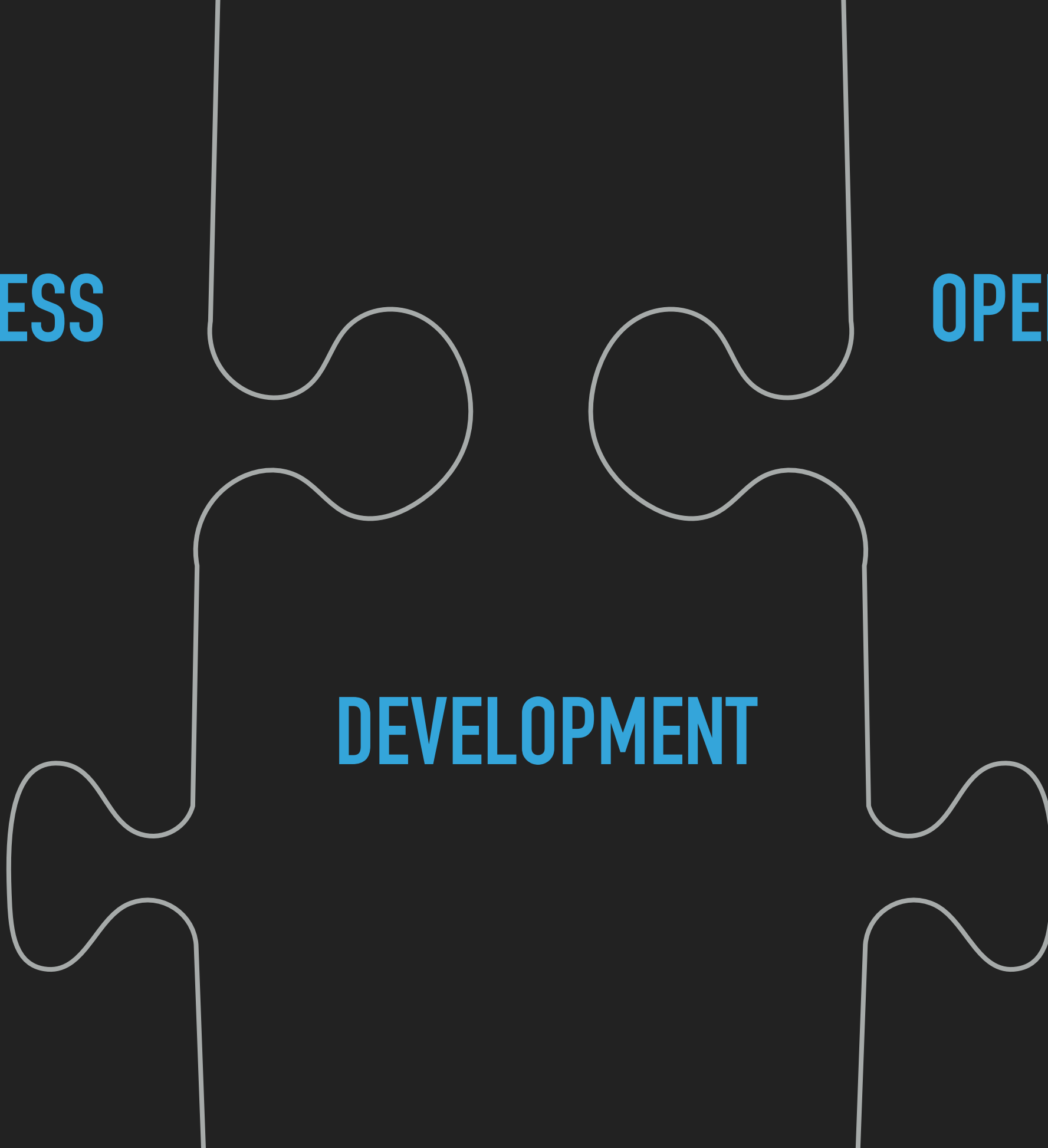
**DEVELOPMENT**



**BUSINESS**

**OPERATIONS**

**DEVELOPMENT**



**BUSINESS**

**OPERATIONS**

**DEVELOPMENT**

**DEVOPS AND AGILE ARE BOTH SYSTEMS THINKING**

---

## AGILE MANIFESTO APPLIES TOO

- ▶ Individuals and interactions over processes and tools
- ▶ Working software over comprehensive documentation
- ▶ Customer collaboration over contract negotiation
- ▶ Responding to change over following a plan



---

## AGILE MANIFESTO APPLIES TOO

- ▶ Individuals and interactions over processes and tools

---

## AGILE MANIFESTO APPLIES TOO

- ▶ Working software over comprehensive documentation

---

## AGILE MANIFESTO APPLIES TOO

- ▶ Customer collaboration over contract negotiation

---

## AGILE MANIFESTO APPLIES TOO

- ▶ Responding to change over following a plan

**BUT HOW?**

KILL THE EXCUSES

**EXCUSES!**

**EXCUSES!**

**(SOME BS, SOME REAL)**

**(SOME BS)**

---

## JEZ HUMBLE: IT WON'T WORK HERE

- ▶ We're regulated
- ▶ We're not building websites
- ▶ Too much legacy
- ▶ Our people are too stupid



---

## JEZ HUMBLE: IT WON'T WORK HERE

- ▶ **We're regulated**

- ▶ Amazon

- ▶ Sarbanes-Oxley and PCI-DSS

- ▶ A deploy to Production  
every second

---

## JEZ HUMBLE: IT WON'T WORK HERE

- ▶ **We're not building websites**

- ▶ Hewlett-Packard

- ▶ Firmware for Printers and Scanners

- ▶ Cost -40%      Productivity +8x

---

## JEZ HUMBLE: IT WON'T WORK HERE

- ▶ **Too much legacy**

- ▶ Suncorp Insurance Agency

- ▶ 15 Green Screen Mainframe Apps

- ▶ Eliminated 14 of 15

---

## JEZ HUMBLE: IT WON'T WORK HERE

- ▶ **Our people are too stupid**
  - ▶ NUMMI - GM + Toyota
    - ▶ GM's worst plant
      - ▶ Add Toyota process -> best plant

**(SOME REAL)**

---

## JEZ HUMBLE: ACTUAL REASONS

- ▶ Our culture sucks
- ▶ Our architecture sucks

---

## JEZ HUMBLE: ACTUAL REASONS

- ▶ Our culture sucks
- ▶ Our architecture sucks

**GO SLOW! TAKE IT EASY.**

# PEOPLE

---

FIND THE VALUE, STOP THE EXCUSES



# PROCESS

---

FIND A BETTER WAY

---

## DEVELOPER MINDSET

- ▶ Decompose Problem
- ▶ Focus on one piece
- ▶ Iterate and Adjust
- ▶ Starting with People and Process

---

# HOLD UP

- ▶ Wait a minute

## OPTIONS FOR CULTURE AND PROCESS

- ▶ Wholesale Culture
- ▶ New Hires
- ▶ < Continuous Integration
- ▶ The Slow Pipeline
- ▶ FOIA Requests
- ▶ Late Automated Feedback
- ▶ Angry Devs and Ops
- ▶ Prod Problems
- ▶ So Many Meetings
- ▶ Adhoc

# WHOLESALE CULTURE CHANGE

---

Crap Culture

**SYSTEMS THINKING  
AMPLIFY FEEDBACK LOOPS  
TESTING AND LEARNING**

Devops Culture

# THE FIX FOR WHOLESALe CULTURE

- ▶ Create the value and the Culture one little step at a time
  - ▶ Systems Thinking
  - ▶ Amplify Feedback Loops
  - ▶ Culture of Continual Experimentation and Learning

# NEW HIRES

---

Person

**A DOCUMENT FOLLOWED BY  
WEEKS OR MONTHS OF  
FINDING OTHER STUFF.**

Employee

## THE FIX

- ▶ Create a new hire feedback loop
- ▶ Define roles and responsibilities as a group



# < CONTINUOUS INTEGRATION

---

Code Changes

**FEATURE BRANCHES.**

**..... MERGE.**

**CODE FREEZE !!**

**PROFIT!!!**

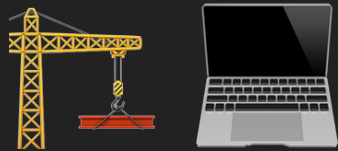
---

## THE FIX

- ▶ Ramp up the continuous!!

# CONTINUOUS INTEGRATION ON \$1 A DAY

# CONTINUOUS INTEGRATION ON \$1 A DAY



\$1

CAVEAT!

---

**BUY IN!**

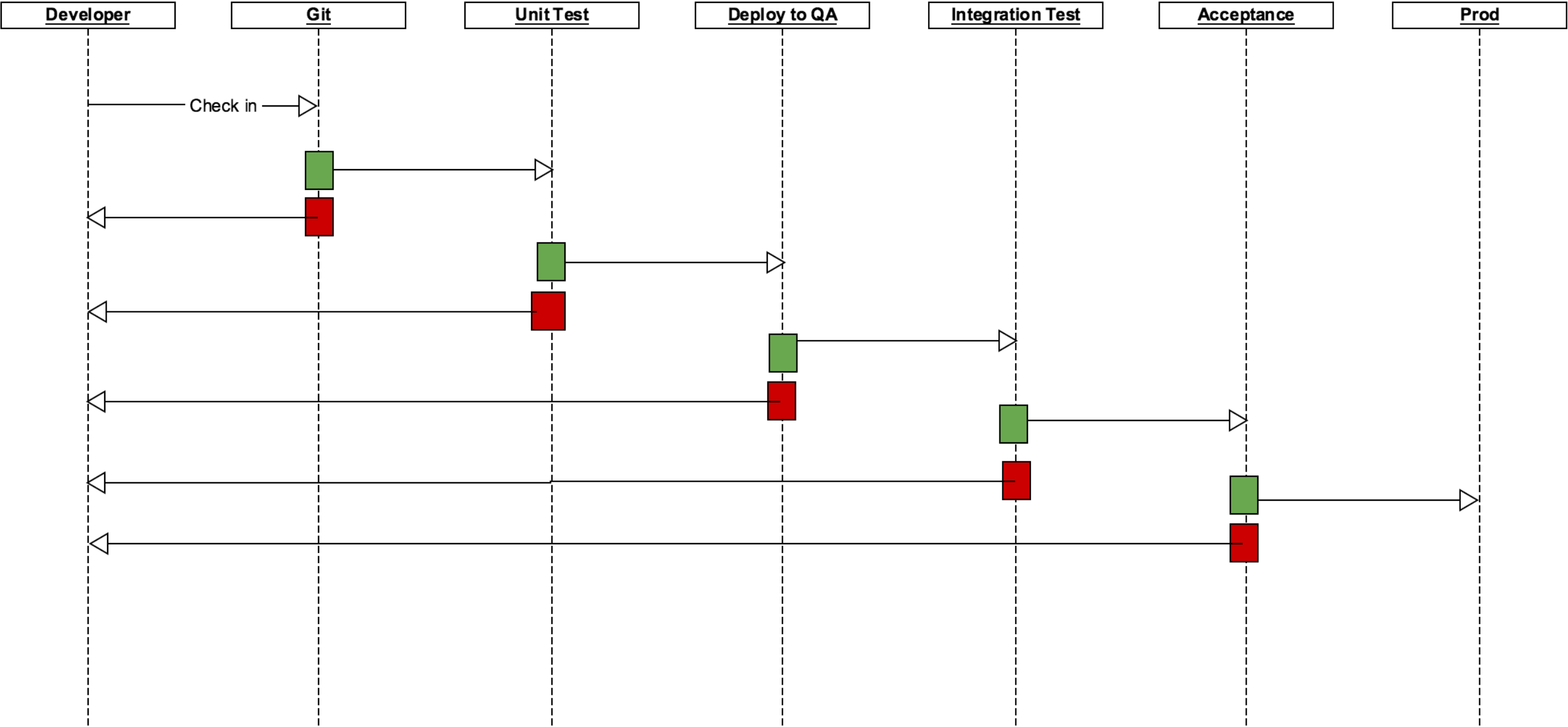
# THE PIPELINE

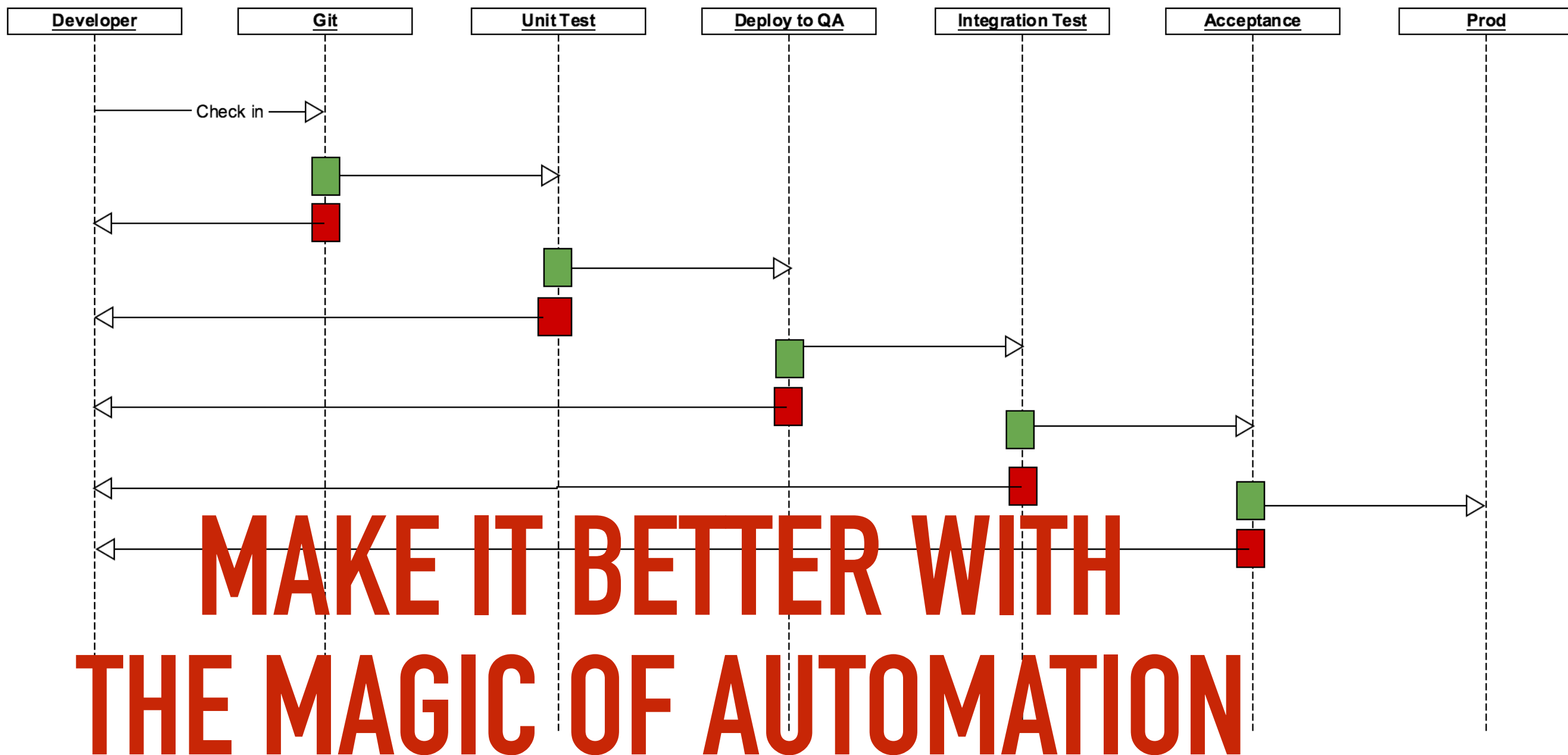
---

Integrated Code Changes

**CODE DEPLOYMENT  
TEST EXECUTION**

Usable Software

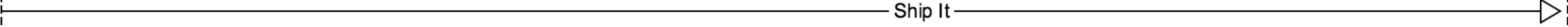
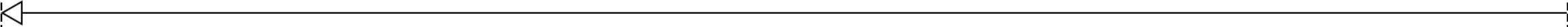






Developer

Prod



# THE SHADOW PIPELINE

---

[Back](#)

Integrated Code Changes

**PRIVATE CODE DEPLOYMENT**  
**PRIVATE TEST EXECUTION**  
**SHHHHHHH**

Usable Software

# FOIA REQUESTS

---

Request for Information

**WORK**

Information

---

# THE FIX

▶ THE SPAM CANNON!

CAVEAT!

---

**HIDDEN VALUE!**

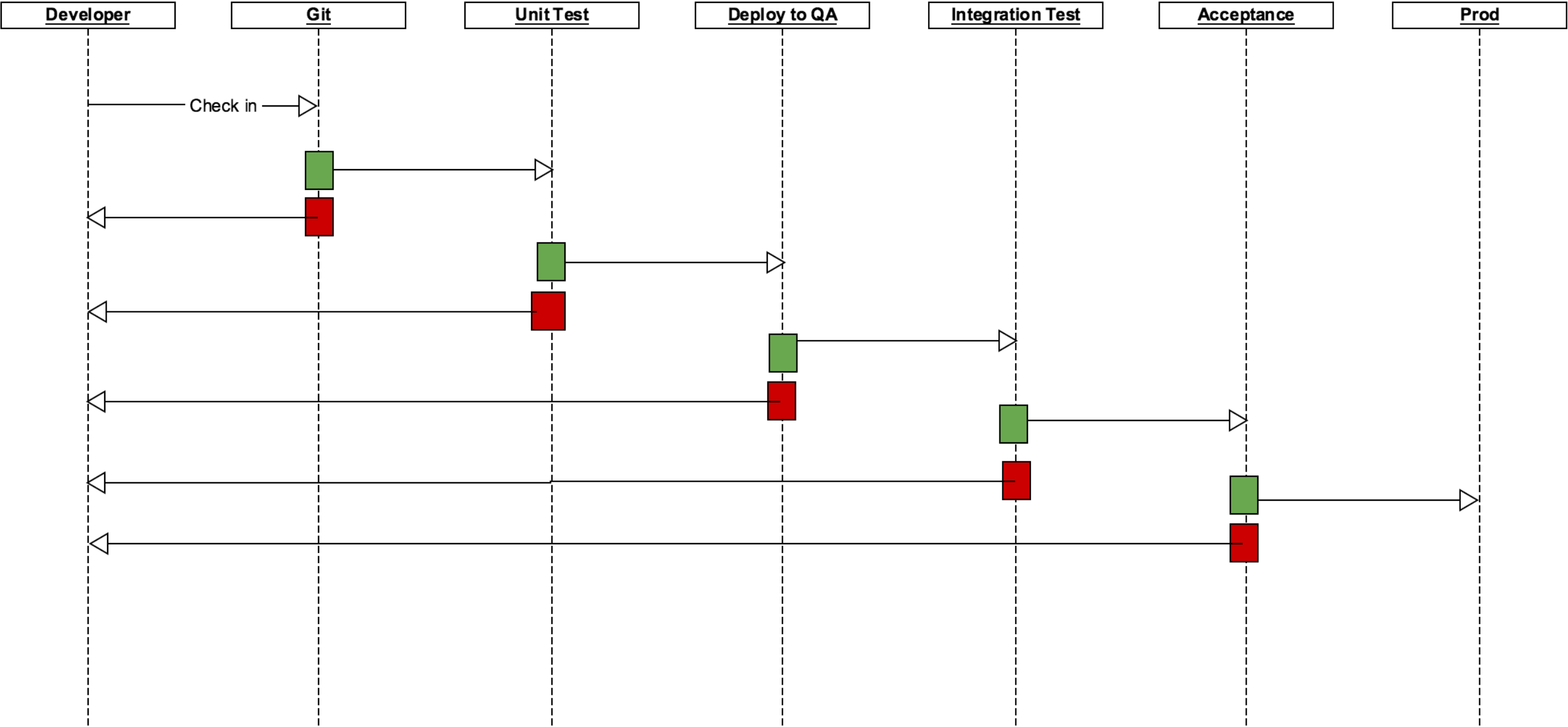
# LATE AUTOMATED FEEDBACK

---

Code into the pipeline

**DAYS OR WEEKS OF  
DEPLOYMENTS AND TESTS**

Errors



CAVEAT!

---

**FIX SOMETHING!**



# ANGRY OPERATIONS AND DEVS

---

Questions and JIRA Tickets

**COMMAND LINE WIZARDRY.  
CAFFEINE CONSUMPTION.  
RAPIDLY MOVING ELECTRONS.**

Growls, stares, and MWYAF

## THE FIX

- ▶ An Olive Branch
  - ▶ Donuts and Coffee
  - ▶ Other Baked Goods
  - ▶ Open Communication

# PROD PROBLEMS

---

Functional code and infrastructure

**BUSINESS  
AS  
USUAL**

Failure

## THE FIX

▶ Start Bridging the Gap

# MEETINGS, MEETINGS, MEETINGS

---

Blood, Sweat, and Tears.

**HOT AIR**

Go / No Go

---

## THE FIX

### ► Gentle Nudges

# THE WORKAROUND

► Break it down

# ADHOC UNIT

---

Input

**WORK**

Output



# THE FIX

- ▶ Break it down

# PROCESS

---

IT MAKES OR BREAKS THE CULTURE

# CULTURE

---

IT MAKES OR BREAKS THE PROCESS

---

## THE FIX FOR WHOLESALe CULTURE

- ▶ Fix Process and the Culture slowly w/
  - ▶ Systems Thinking
  - ▶ Amplify Feedback Loops
  - ▶ Culture of Continual Experimentation and Learning

# ARCHITECTURE

---

MAKE A BETTER WAY

---

## JEZ HUMBLE: ACTUAL REASONS

- ▶ Our culture sucks
- ▶ **Our architecture sucks**



A LOUDER WAY

---

**THE BIG BANG**





A BETTER WAY

---

**THE STRANGLER  
APPLICATION**



# ARCHITECTURE

---

MADE A BETTER WAY, SLOWLY

# QUESTIONS?

---

ANSWERS: CHECK EM OUT

- ▶ Rob Kinyon: <https://streamlined-book.blogspot.com/>
- ▶ Jez Humble: <https://continuousdelivery.com/>
- ▶ Gene Kim: <https://itrevolution.com/>
- ▶ Damian Synadinos: <http://www.ineffable-solutions.com/>
- ▶ James Shore: <http://www.jamesshore.com/Blog/>
- ▶ Richard Bradshaw: <http://www.thefriendlytester.co.uk/>
- ▶ STeve Shary: <http://stevesmashcode.com/>
- ▶ Martin Fowler: <https://www.martinfowler.com/>
- ▶ <https://www.thisamericanlife.org/radio-archives/episode/561/nummi-2015>
- ▶ craigles75: <https://flic.kr/p/N5mcKY>
- ▶ Me, Slim, Jason Blackhurst: [@alsoKnownAsSlim](mailto:JasonBlackhurst@gmail.com)
- ▶ License: [Creative Commons Attribution ShareAlike 3.0](#)

## SUM UP

- ▶ Use the Developer Mindset
  - ▶ Decompose
  - ▶ Iterate on one piece
  - ▶ Make Feedback Fast
- ▶ I wish you the Best of Luck! Thank You!
- ▶ [JasonBlackhurst@gmail.com](mailto:JasonBlackhurst@gmail.com) ----  
@AlsoKnownAsSlim