Why Can’t We CD?

The common issues that stop focus, flow, and joy

Walmart Global Tech

Bryan Finster

Staff Software Engineer
Value Stream Architect
Walmart DevOps Dojo

Why
Continuous Delivery?

Legacy Delivery

1. Build everything to spec

Legacy Delivery

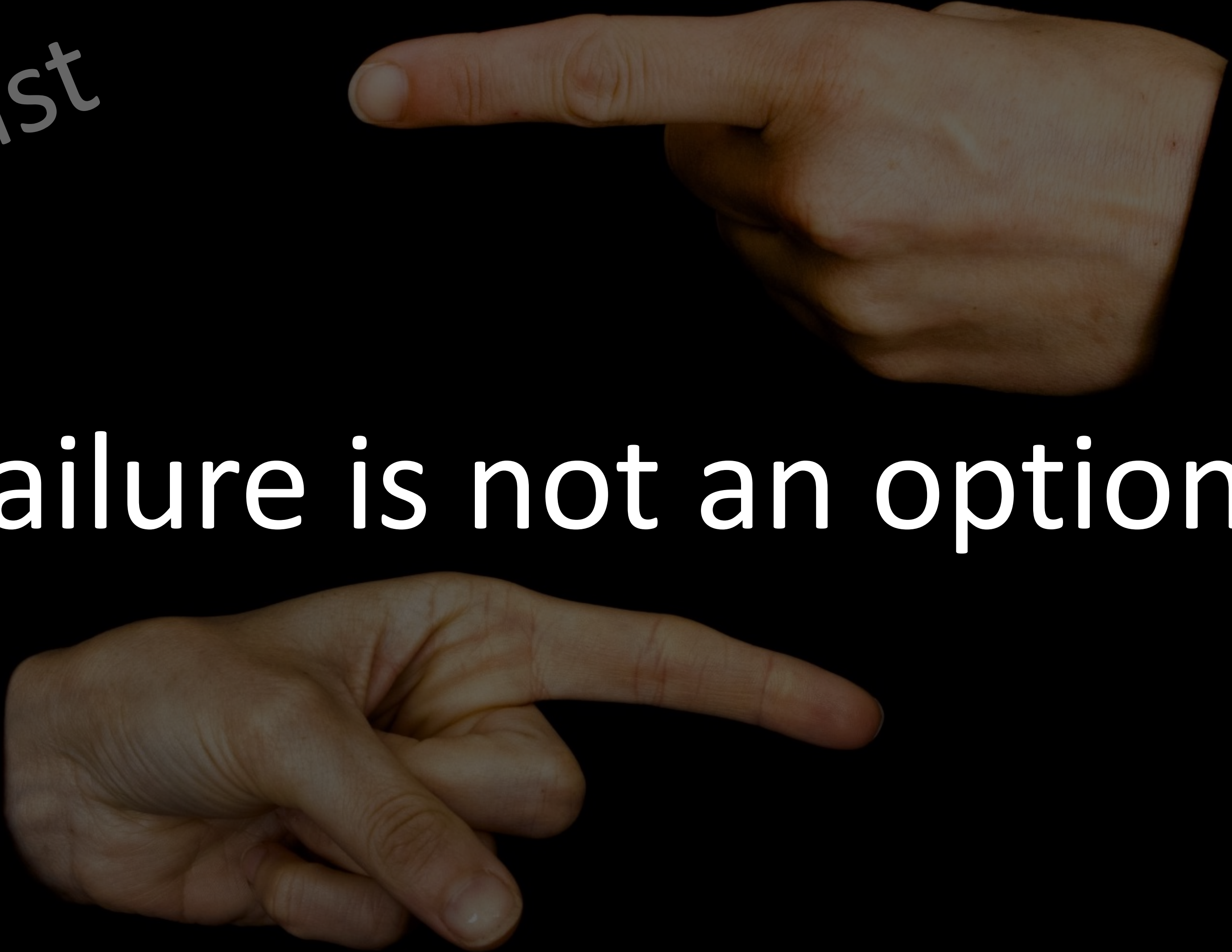
1. Build everything to spec
2. Hand it off to test

Legacy Delivery

1. Build everything to spec
2. Hand it off to test
3. Deliver it

Legacy Delivery

1. Build everything to spec
2. Hand it off to test
3. Deliver it
4. War room



Blame

Mistrust

Fear

"Failure is not an option!"

Why We Fail

The requirements are wrong

We misunderstand them

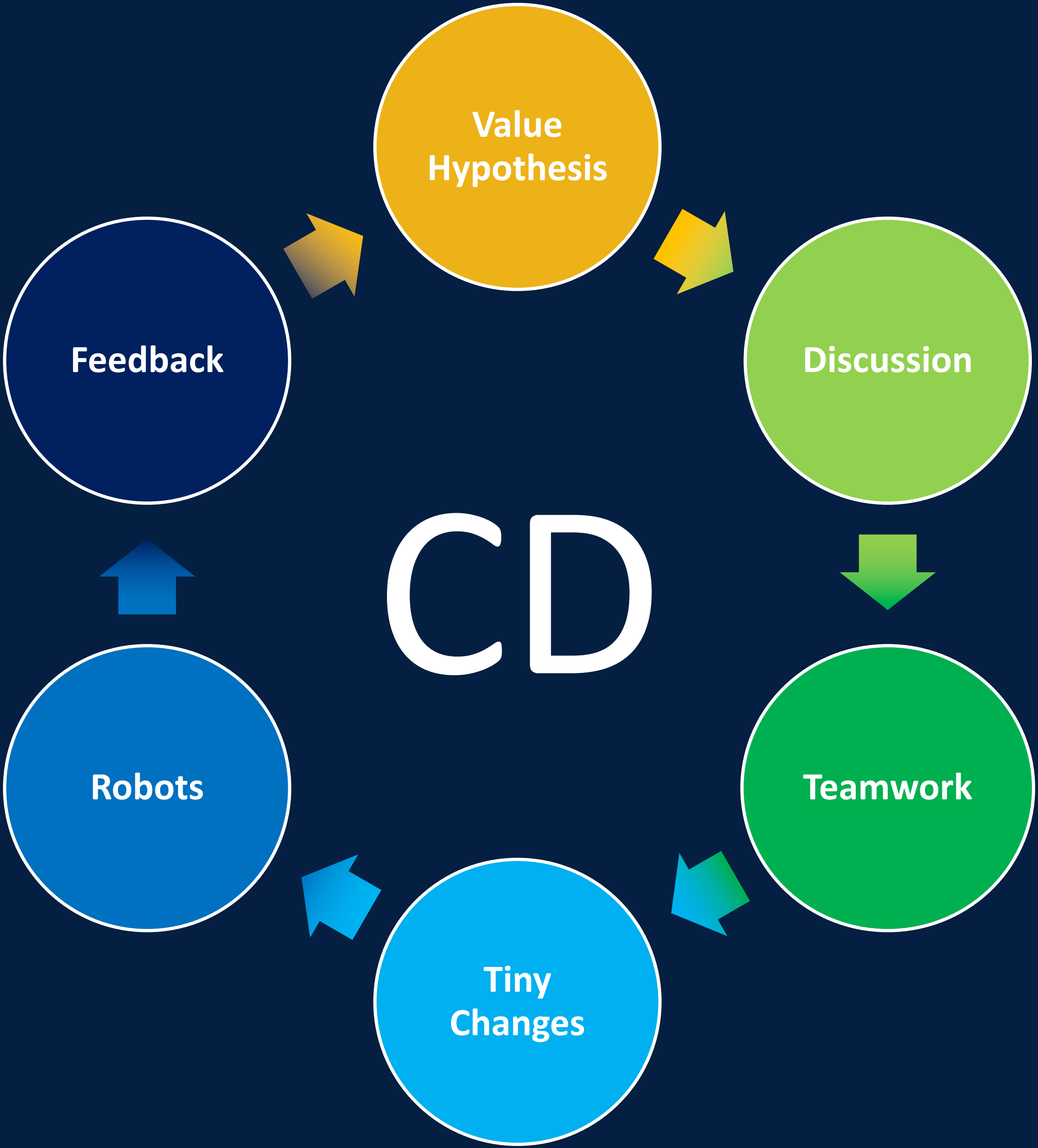
They change rapidly

Failure is a promise

Preparing is optional

“We automated the build!”
“We’re doing CD!
“Why didn’t things get things better?!”

CD is a practice, not robots



Faster feedback

Sooner, safer value delivery

More humane way of working

The Main Challenges

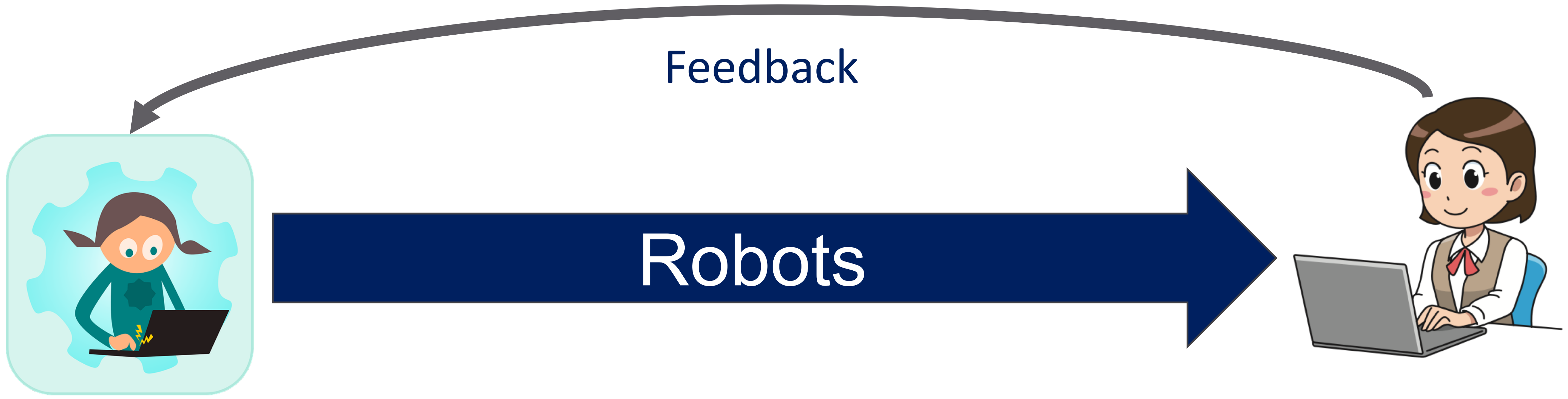
1. Leaky Pipelines

Handoffs

"Hot fix" exception flows

Manual processes

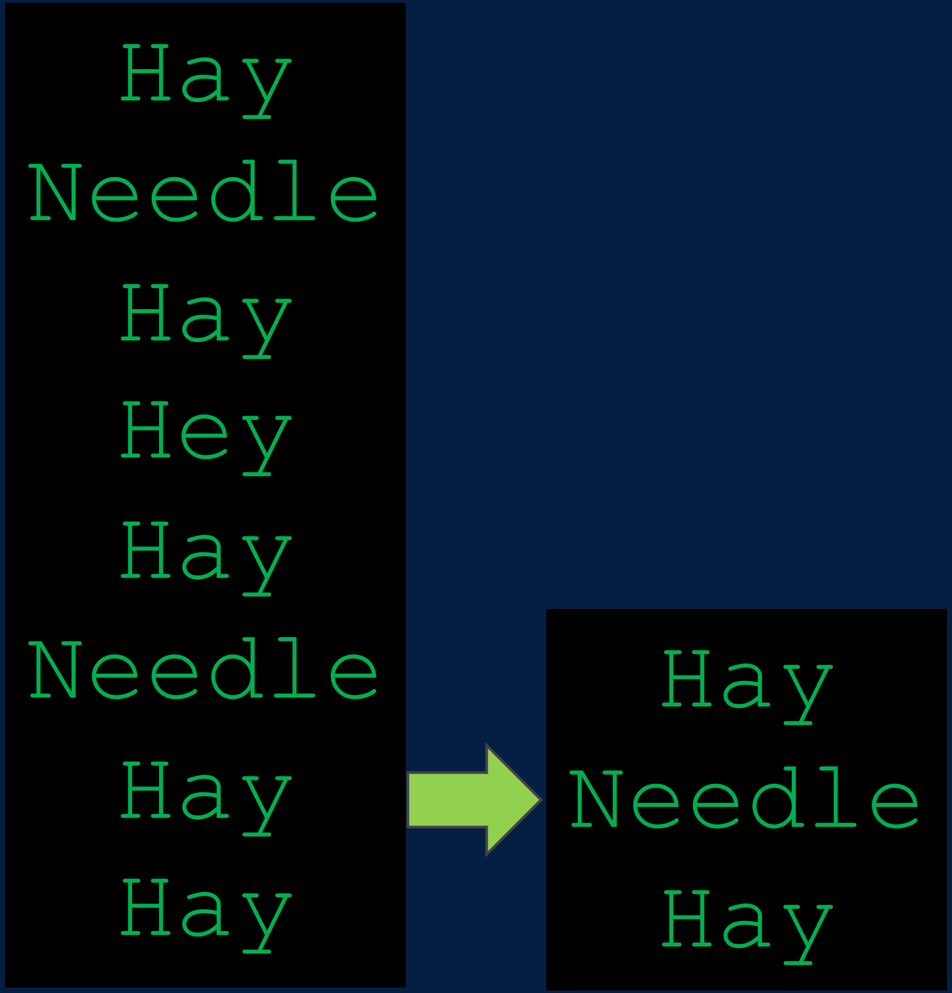
Single Automated Path to Deliver



Establishing an automated feedback loop is the highest priority…

and 10% of the solution

Automated Feedback



Enables smaller batches

Easier to find needles

Reduces waste & toil

Only if we make small changes

2. Eventual Integration

“Why would we merge before it’s complete?”

Quality requires

Continuous Integration

Global Tech

Large Batches Create Needles

```
  Hay
Needle
  Hay
  Hey
  Hay
Needle
  Hay
  Hay
```

Waiting for “feature complete”

Heavy code review process

External approvals

Ineffective & manual testing

CI Removes Needles

All changes integrated to trunk at least daily

~80% confident before commit

Feedback from trunk in < 10 minutes

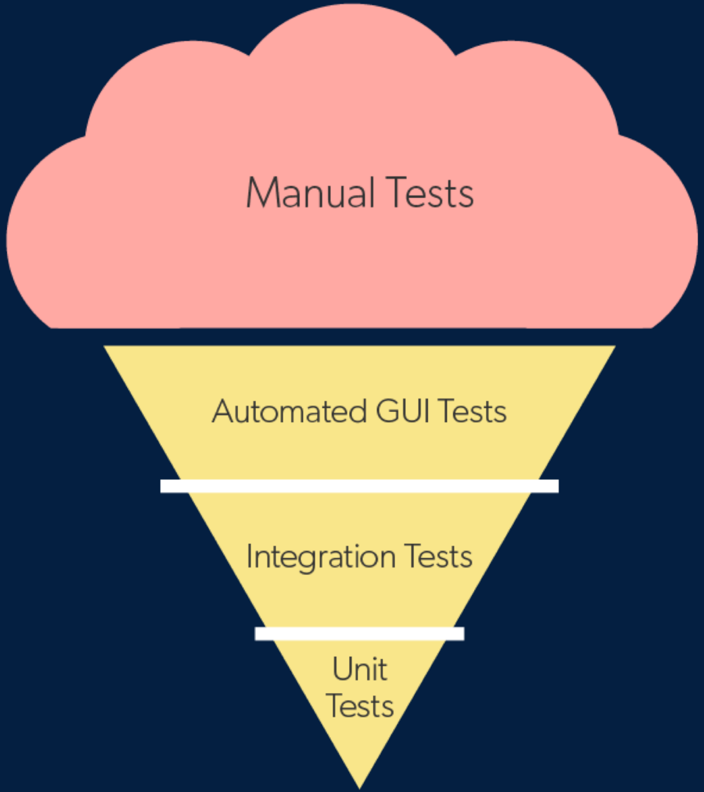
CI improves quality:

Uncovers upstream waste and quality issues

3. Ineffective Test Architecture

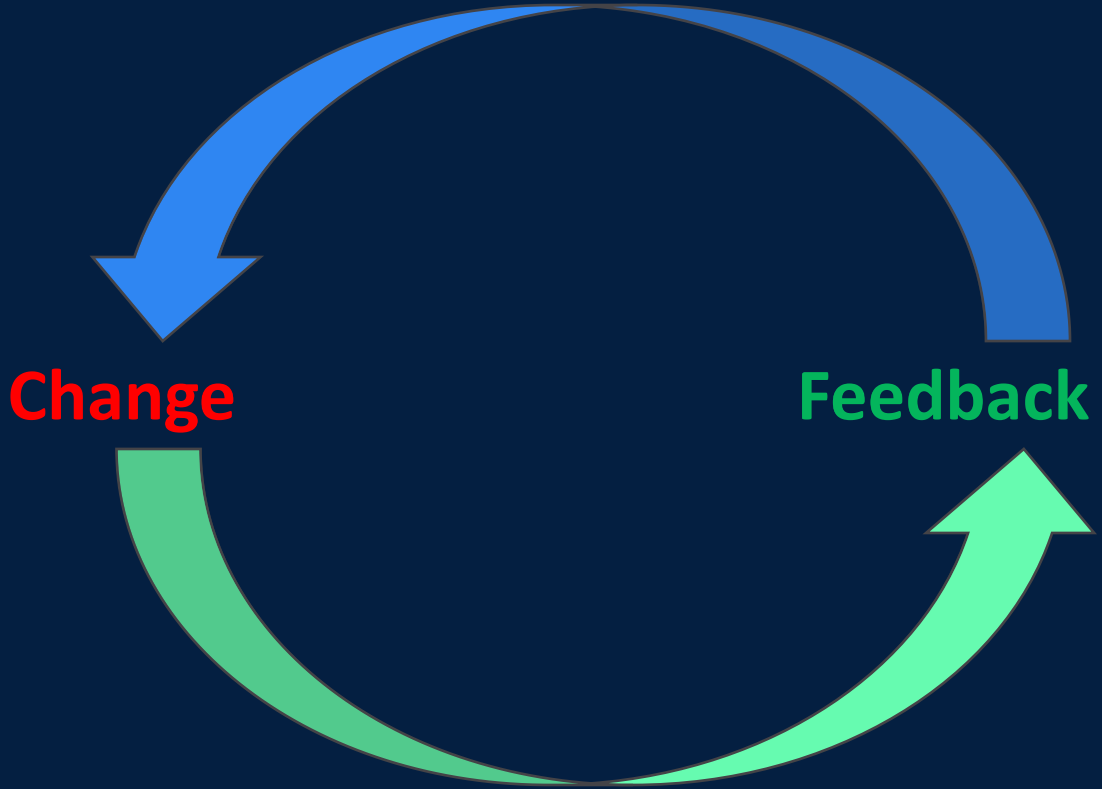
Global Tech

Confident before commit?



Not with this test architecture

Efficient & Effective Tests



Reliably report poor quality

Execute quickly

Alert early in the flow

Optimize for quality feedback

Feedback Blockers

Infrequent delivery

Feedback Blockers

Infrequent delivery

Testing as a secondary activity

Feedback Blockers

Infrequent delivery

Testing as a secondary activity

Flaky tests

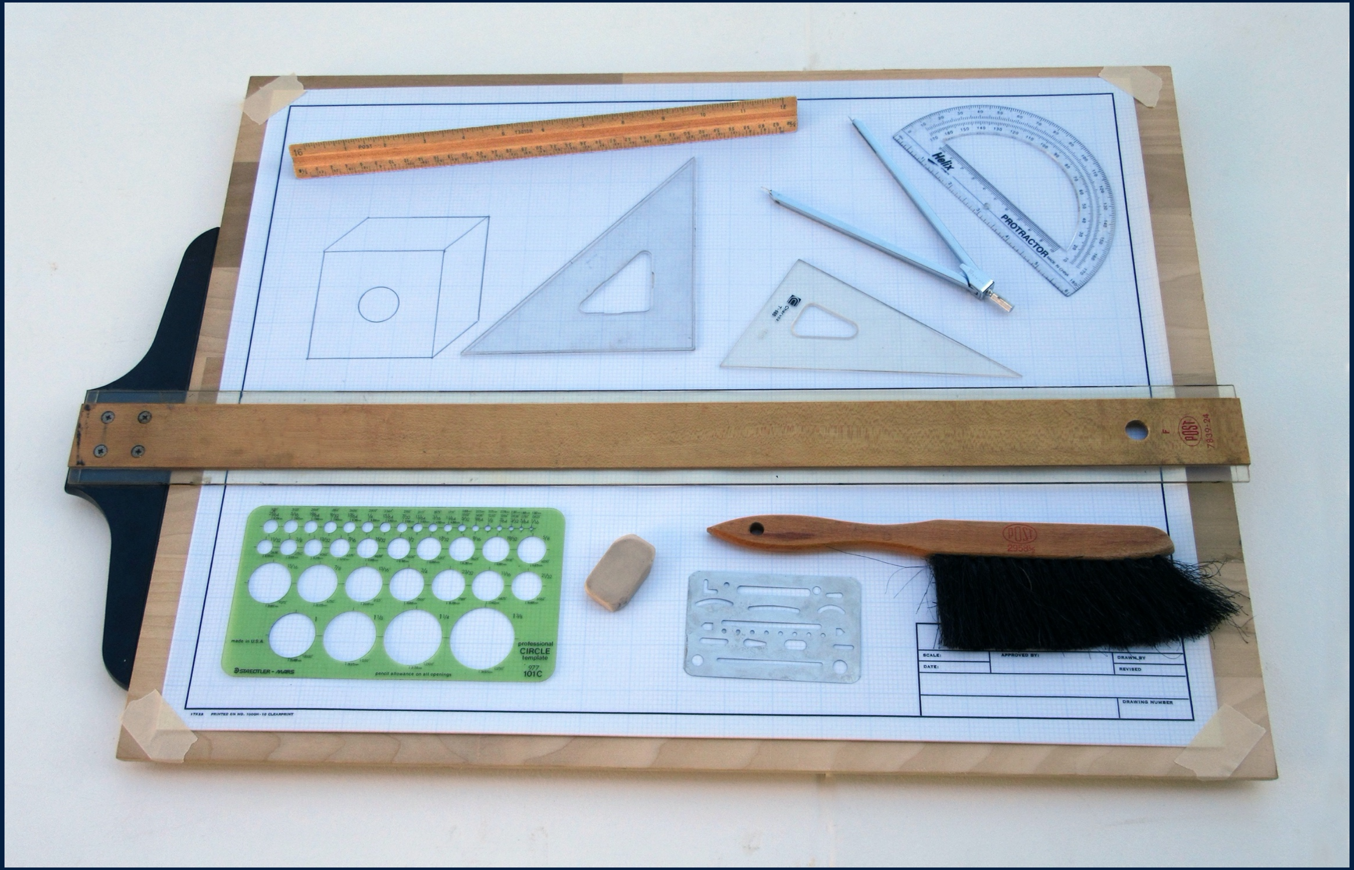
Feedback Blockers

Infrequent delivery

Testing as a secondary activity

Flaky tests

Poor test design



Feedback Blockers

Infrequent delivery

Testing as a secondary activity

Flaky tests

Poor test design

Vague requirements

GIGO

4. Vague Requirements

“Just do what I mean”

”Figure it out during development”

Lack of Direction

Most defects are created before coding starts

Behavior Driven Development

Clear business goals

Described with testable outcomes

Reduces batch size

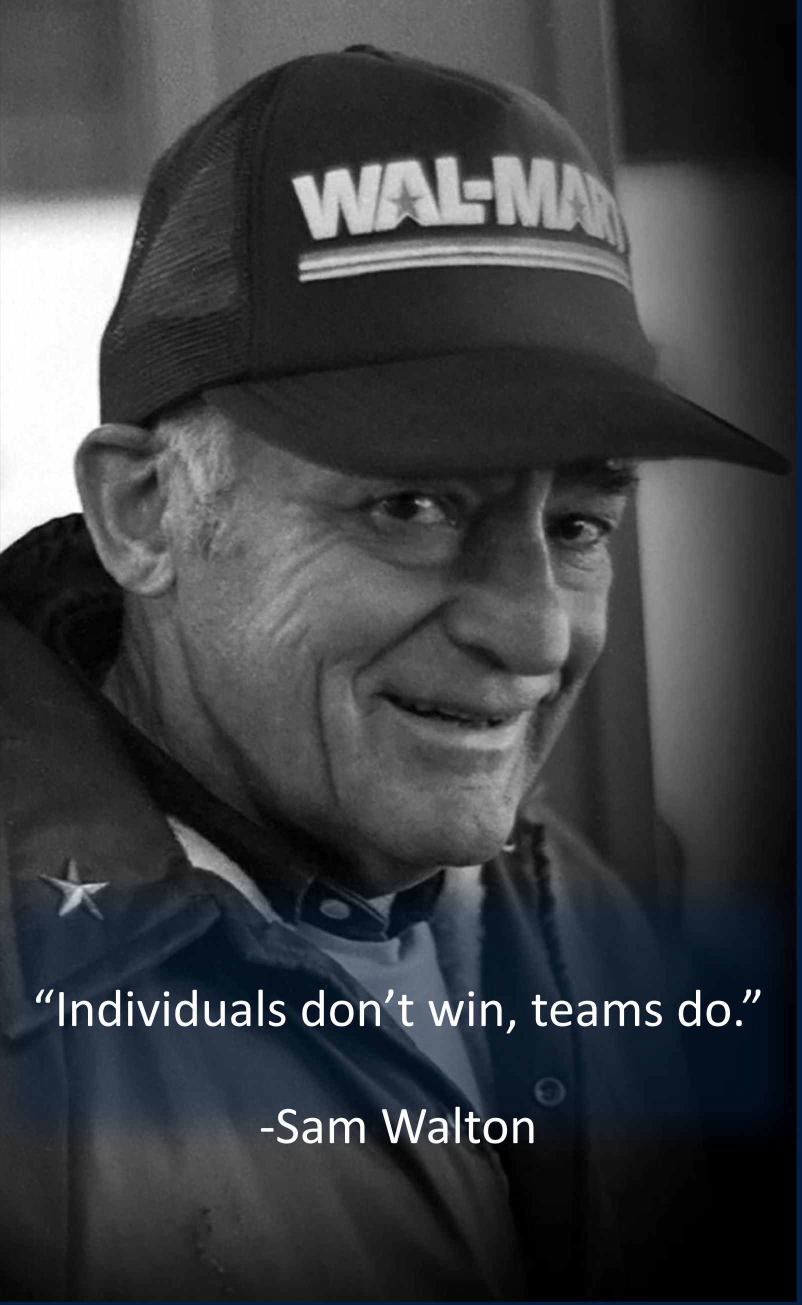
So we can get feedback and adjust

```
As an hourly associate I want to log my arrival time
so that I can be paid correctly.

Scenario: I am not clocked in
  Given I am not clocked in
  When I enter my associate number
  Then my arrival time will be logged
  And I will be notified of the time

Scenario: I clocked in more than 5 minutes ago
  Given I am clocked in
  When I enter my associate number
  And I have been clocked in for more than 5 minutes
  Then I will be clocked out
  And I will be notified of the time

~~Scenario: I clocked in less than 5 minutes ago~~
~~  Given I am clocked in~~
~~  When I enter my associate number~~
~~  And I have been clocked in for less than 5 minutes~~
~~  Then I will receive an error~~
```

Quality Requires Ownership

Own·er·ship: \ˈō-nər-ˌship\ - noun

Having clear goals,
the tools to accomplish them,
responsibility for the outcomes,
and authority to improve them.



"Individuals don't win, teams do."

-Sam Walton

Optimize for Value Delivery

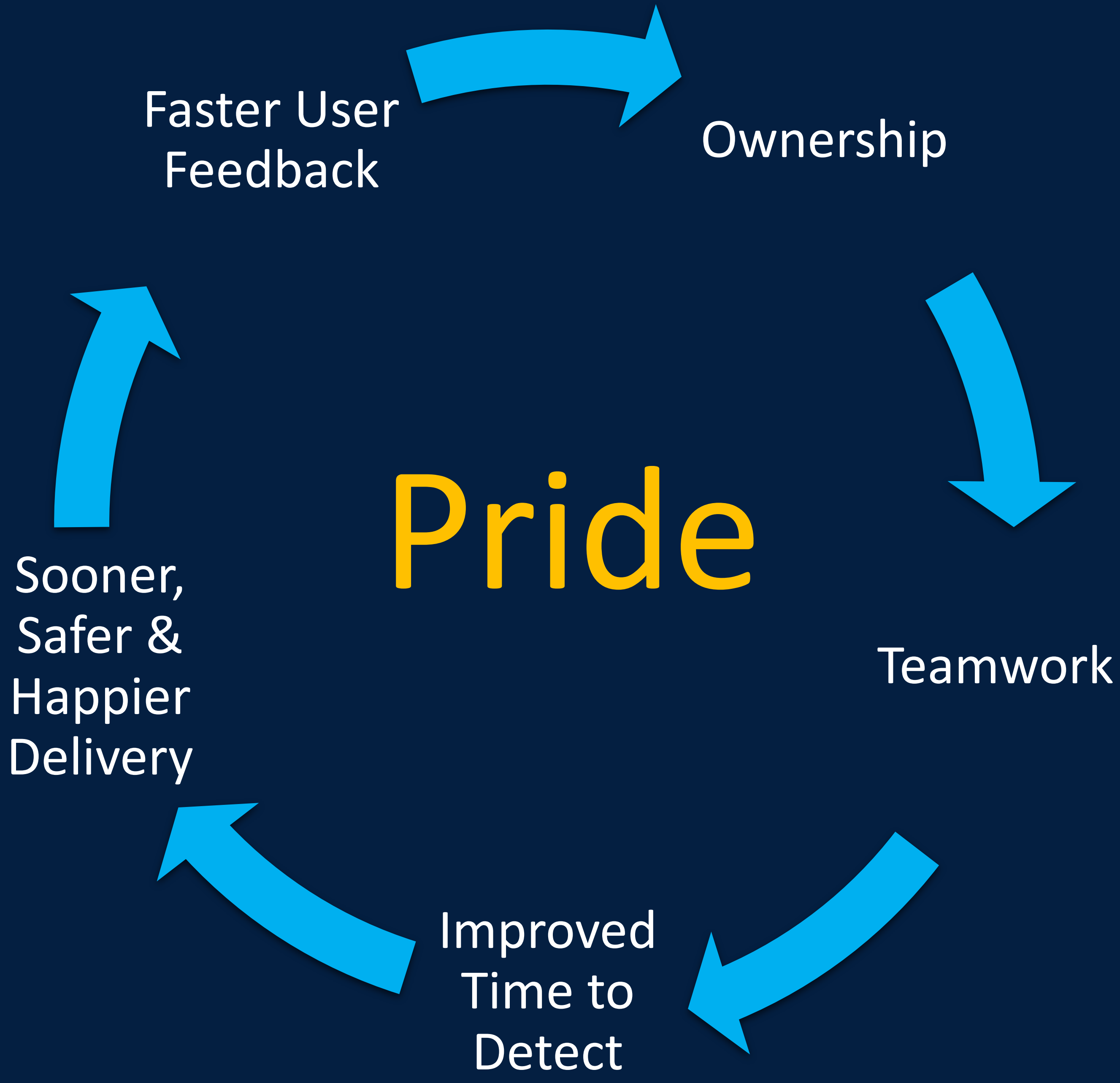
Build a pipeline to get quality feedback.

Relentlessly improve that feedback.

Organize around communication.

If our goal is delivering value…

We'll fix any organizational issues that prevent this.


Faster User Feedback
Ownership
Pride
Teamwork
Improved Time to Detect
Sooner, Safer & Happier Delivery

Outcomes
Crush
Output

Productivity is not how much we deliver.

Productivity is how well we support the customer.

Thank you!

LinkedIn: bryan-finster

Medium: bdfinst.medium.com

Twitter: @BryanFinster