ThoughtWorks®

ACTIONABLE CONTINUOUS DELIVERY METRICS

Suzanne Prince, Head of Product, ThoughtWorks Products

Head of Product for ThoughtWorks Products

13+ years experience with agile, continuous integration and continuous delivery practices

Tweeting @pm_suzie

ThoughtWorks®

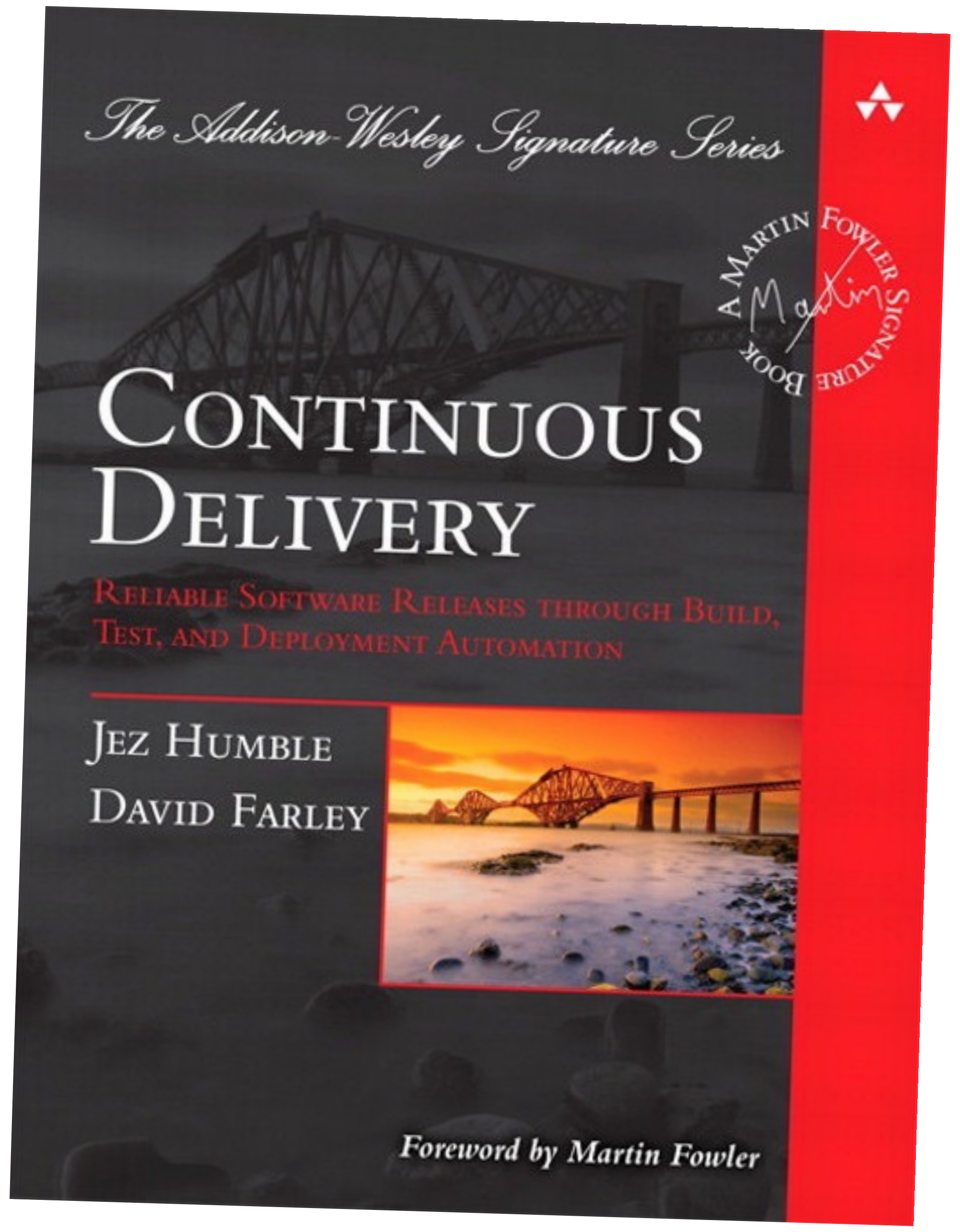
over 20 years

4000 people

40 offices

14 countries

ThoughtWorks®
PRODUCTS


The Addison-Wesley Signature Series

A Martin Fowler Signature Book

Continuous Delivery

Reliable Software Releases through Build, Test, and Deployment Automation

Jez Humble
David Farley

Foreword by Martin Fowler

cruisecontrol™

Se

go

THIS TALK

THIS TALK

- What is continuous delivery (CD)

THIS TALK

- What is continuous delivery (CD)
- Explain why you should measure your continuous delivery process

THIS TALK

- What is continuous delivery (CD)
- Explain why you should measure your continuous delivery process
- Share what continuous delivery metrics you should measure

THIS TALK

- What is continuous delivery (CD)
- Explain why you should measure your continuous delivery process
- Share what continuous delivery metrics you should measure
- Review some scenarios to explain what certain metrics reveal about your continuous delivery process

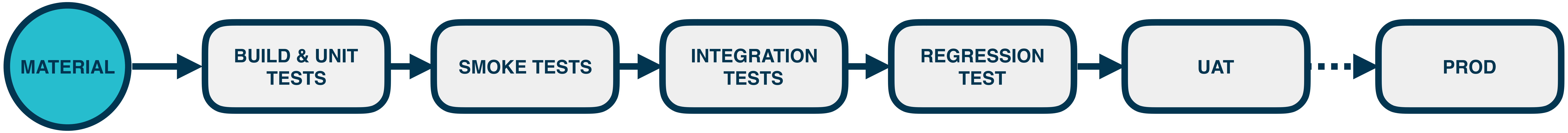
THIS TALK

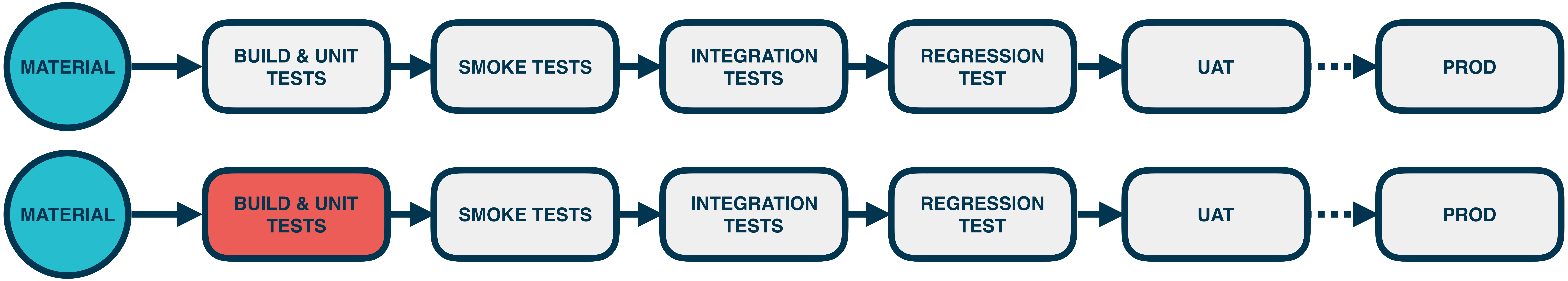
- What is continuous delivery (CD)
- Explain why you should measure your continuous delivery process
- Share what continuous delivery metrics you should measure
- Review some scenarios to explain what certain metrics reveal about your continuous delivery process
- Questions

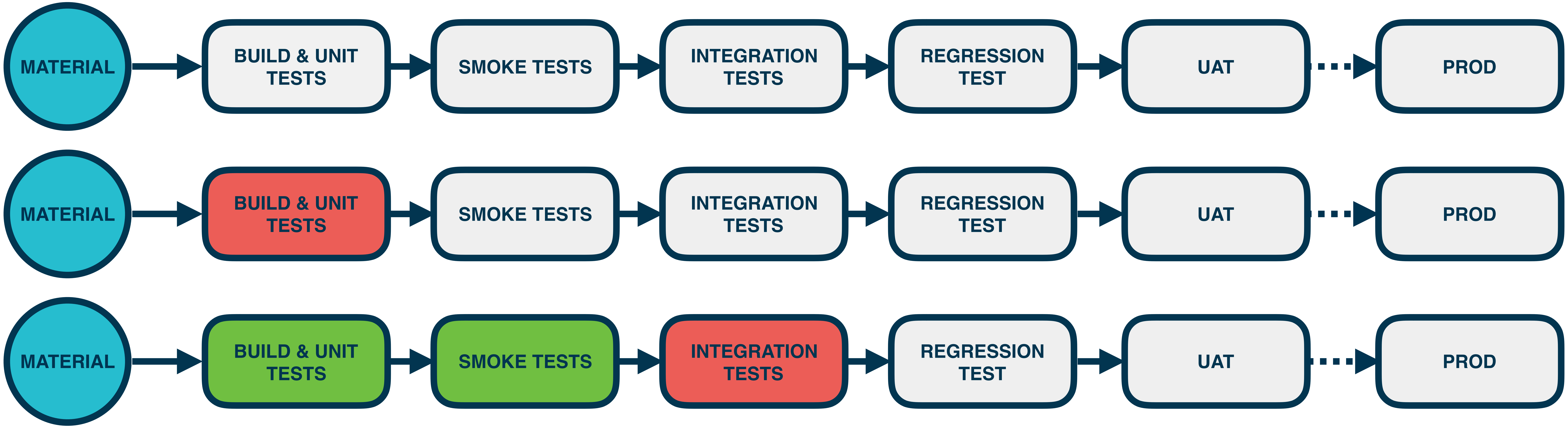
CONTINUOUS DELIVERY

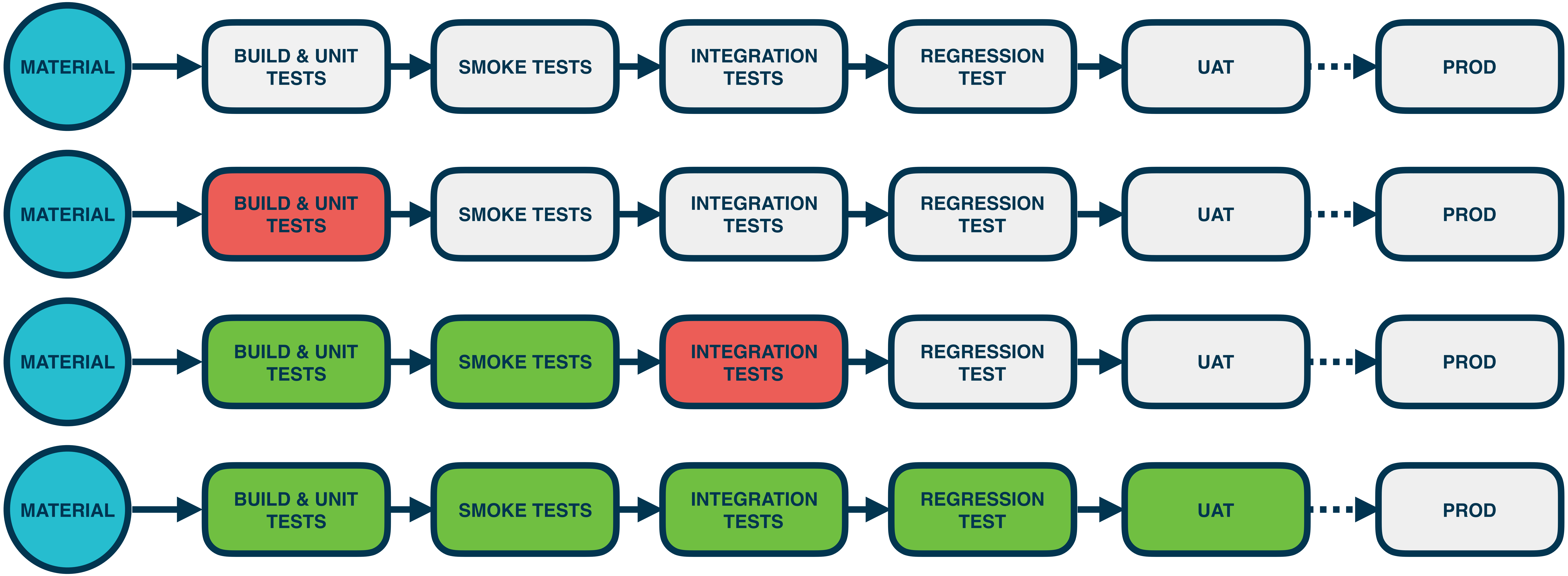
“Continuous Delivery is the ability to get changes of all types—including new features, configuration changes, bug fixes and experiments—into production, or into the hands of users, safely and quickly in a sustainable way.”

- Jez Humble, continuousdelivery.com


MATERIAL
BUILD & UNIT TESTS
SMOKE TESTS
INTEGRATION TESTS
REGRESSION TEST
UAT
PROD


MATERIAL
BUILD & UNIT TESTS
SMOKE TESTS
INTEGRATION TESTS
REGRESSION TEST
UAT
PROD
MATERIAL
BUILD & UNIT TESTS
SMOKE TESTS
INTEGRATION TESTS
REGRESSION TEST
UAT
PROD


MATERIAL
BUILD & UNIT TESTS
SMOKE TESTS
INTEGRATION TESTS
REGRESSION TEST
UAT
PROD
MATERIAL
BUILD & UNIT TESTS
SMOKE TESTS
INTEGRATION TESTS
REGRESSION TEST
UAT
PROD
MATERIAL
BUILD & UNIT TESTS
SMOKE TESTS
INTEGRATION TESTS
REGRESSION TEST
UAT
PROD

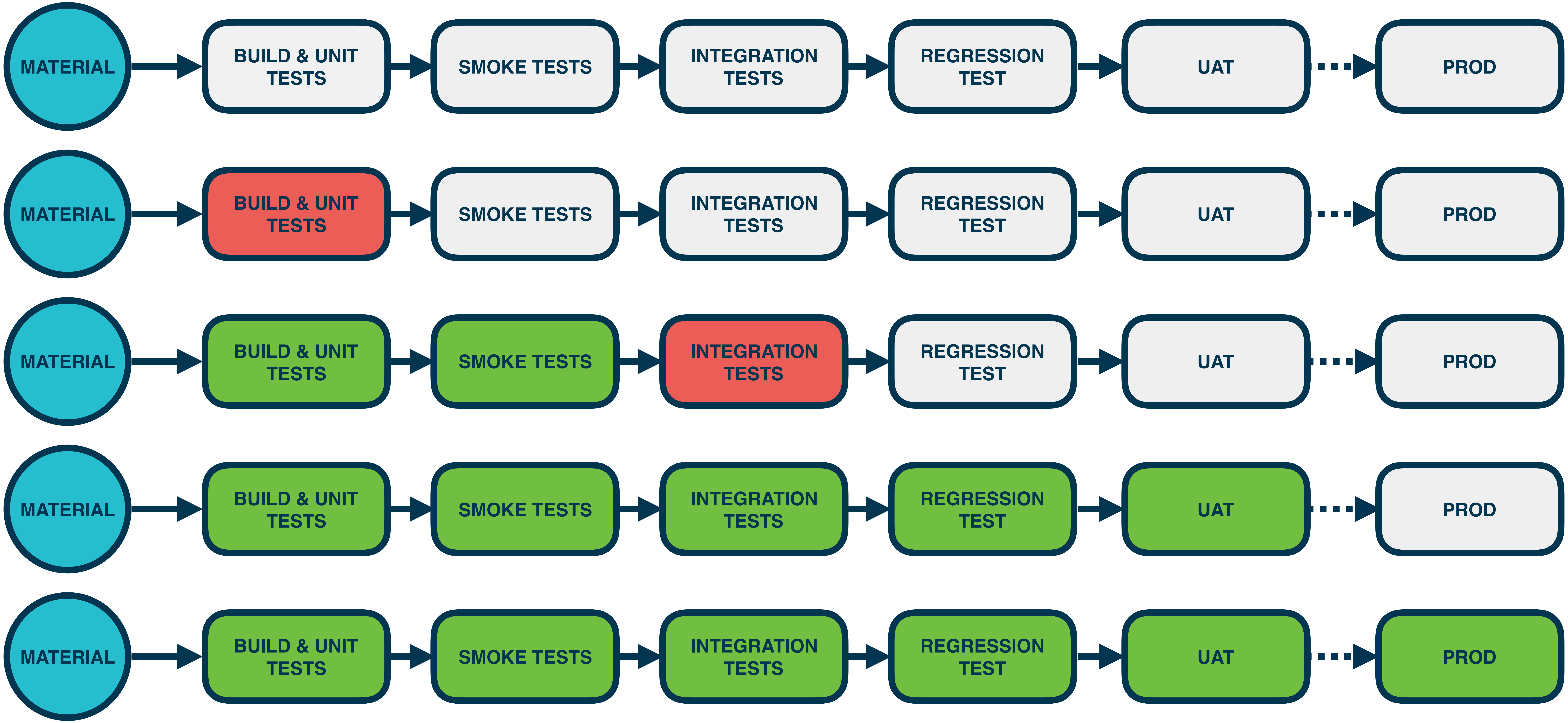


MATERIAL → BUILD & UNIT TESTS → SMOKE TESTS → INTEGRATION TESTS → REGRESSION TEST → UAT ⋯→ PROD

MATERIAL → BUILD & UNIT TESTS → SMOKE TESTS → INTEGRATION TESTS → REGRESSION TEST → UAT ⋯→ PROD

MATERIAL → BUILD & UNIT TESTS → SMOKE TESTS → INTEGRATION TESTS → REGRESSION TEST → UAT ⋯→ PROD

MATERIAL → BUILD & UNIT TESTS → SMOKE TESTS → INTEGRATION TESTS → REGRESSION TEST → UAT ⋯→ PROD



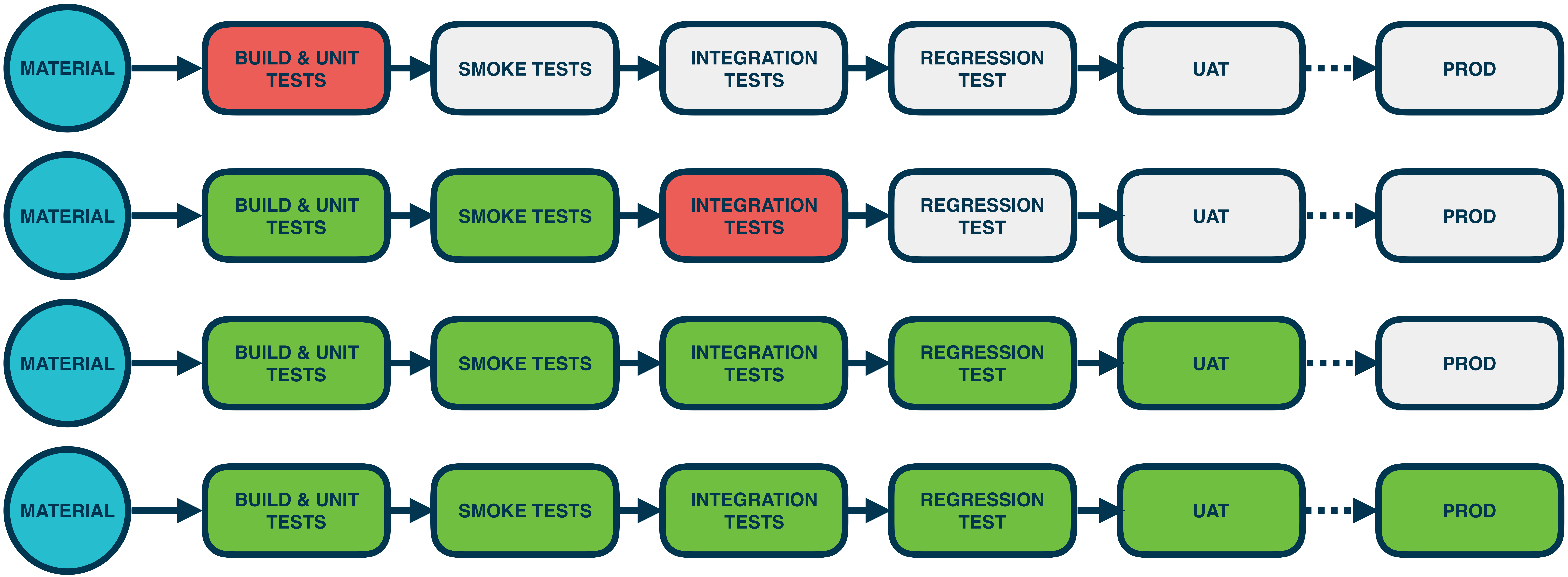
MATERIAL → BUILD & UNIT TESTS → SMOKE TESTS → INTEGRATION TESTS → REGRESSION TEST → UAT ··· → PROD

MATERIAL → BUILD & UNIT TESTS → SMOKE TESTS → INTEGRATION TESTS → REGRESSION TEST → UAT ··· → PROD

MATERIAL → BUILD & UNIT TESTS → SMOKE TESTS → INTEGRATION TESTS → REGRESSION TEST → UAT ··· → PROD

MATERIAL → BUILD & UNIT TESTS → SMOKE TESTS → INTEGRATION TESTS → REGRESSION TEST → UAT ··· → PROD

MATERIAL → BUILD & UNIT TESTS → SMOKE TESTS → INTEGRATION TESTS → REGRESSION TEST → UAT ··· → PROD

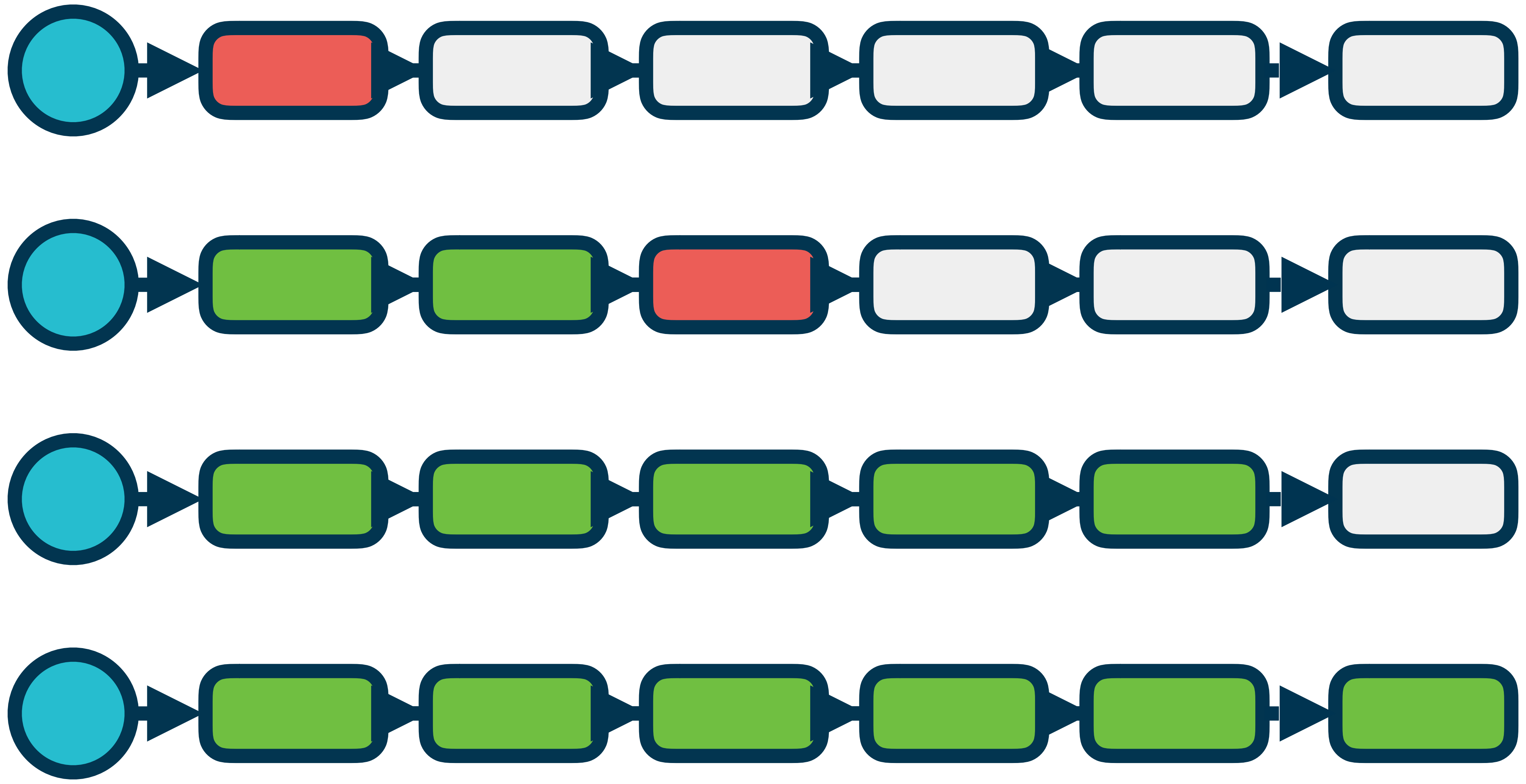


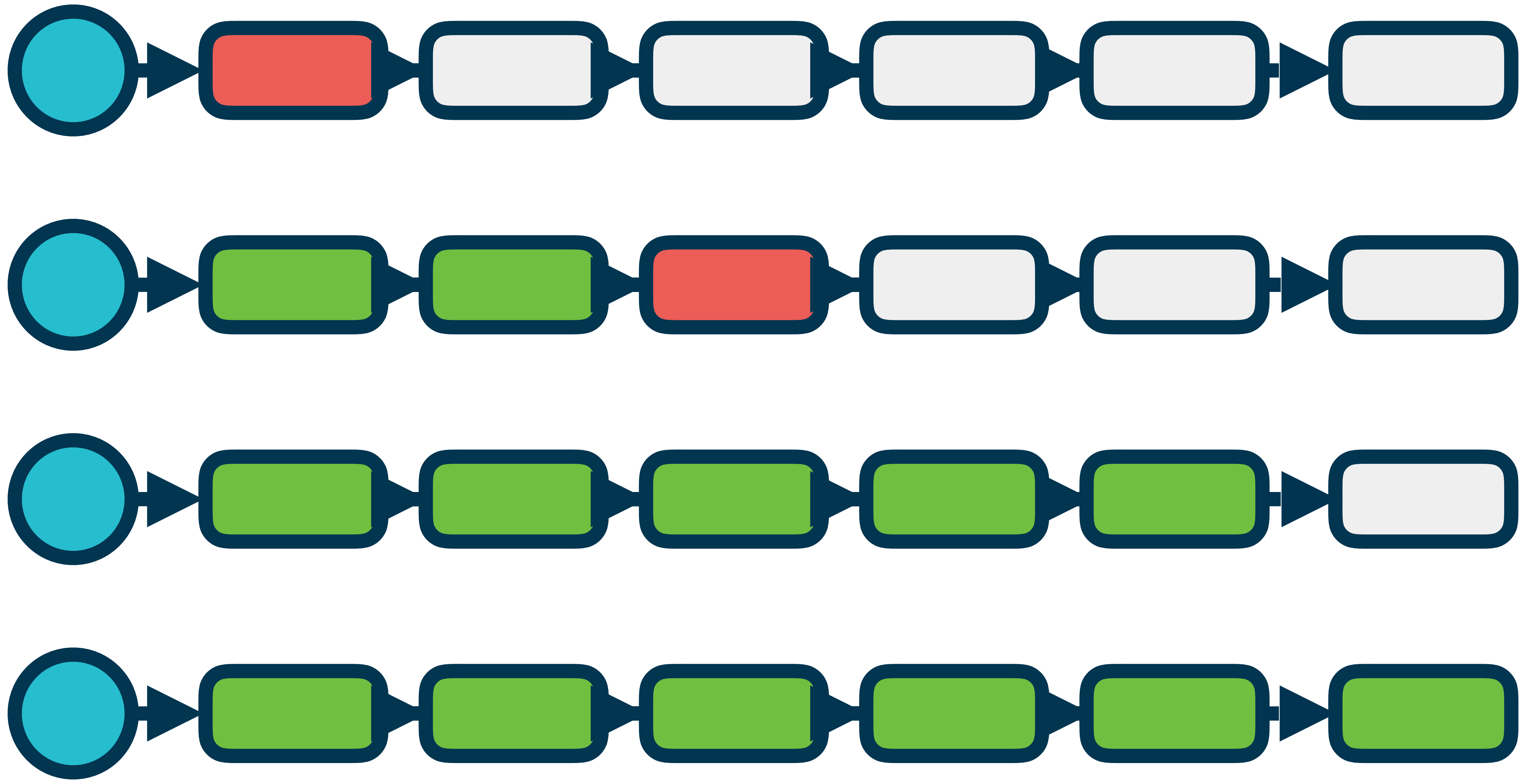
MATERIAL → BUILD & UNIT TESTS → SMOKE TESTS → INTEGRATION TESTS → REGRESSION TEST → UAT → PROD

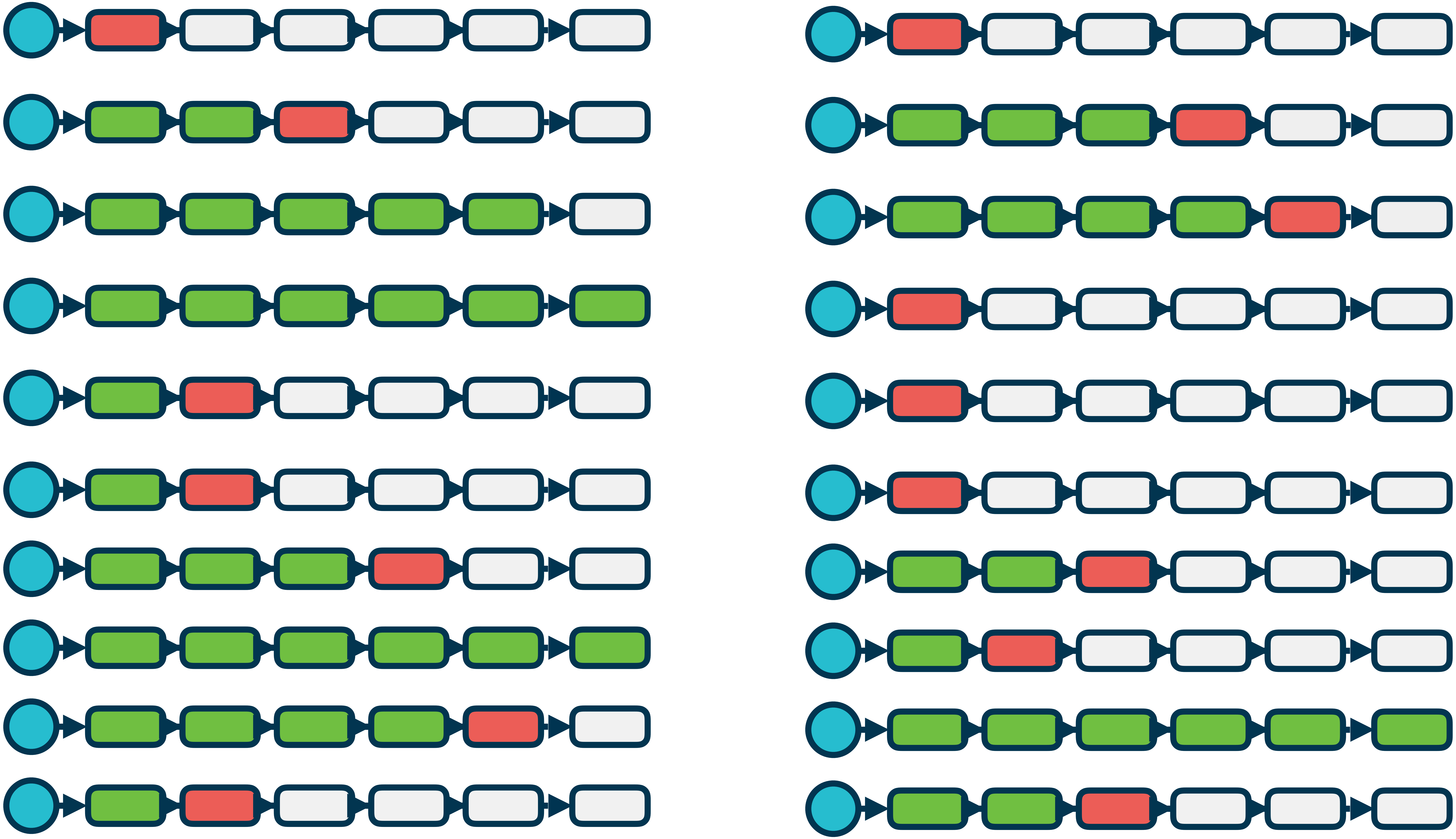
MATERIAL → BUILD & UNIT TESTS → SMOKE TESTS → INTEGRATION TESTS → REGRESSION TEST → UAT → PROD

MATERIAL → BUILD & UNIT TESTS → SMOKE TESTS → INTEGRATION TESTS → REGRESSION TEST → UAT → PROD

MATERIAL → BUILD & UNIT TESTS → SMOKE TESTS → INTEGRATION TESTS → REGRESSION TEST → UAT → PROD

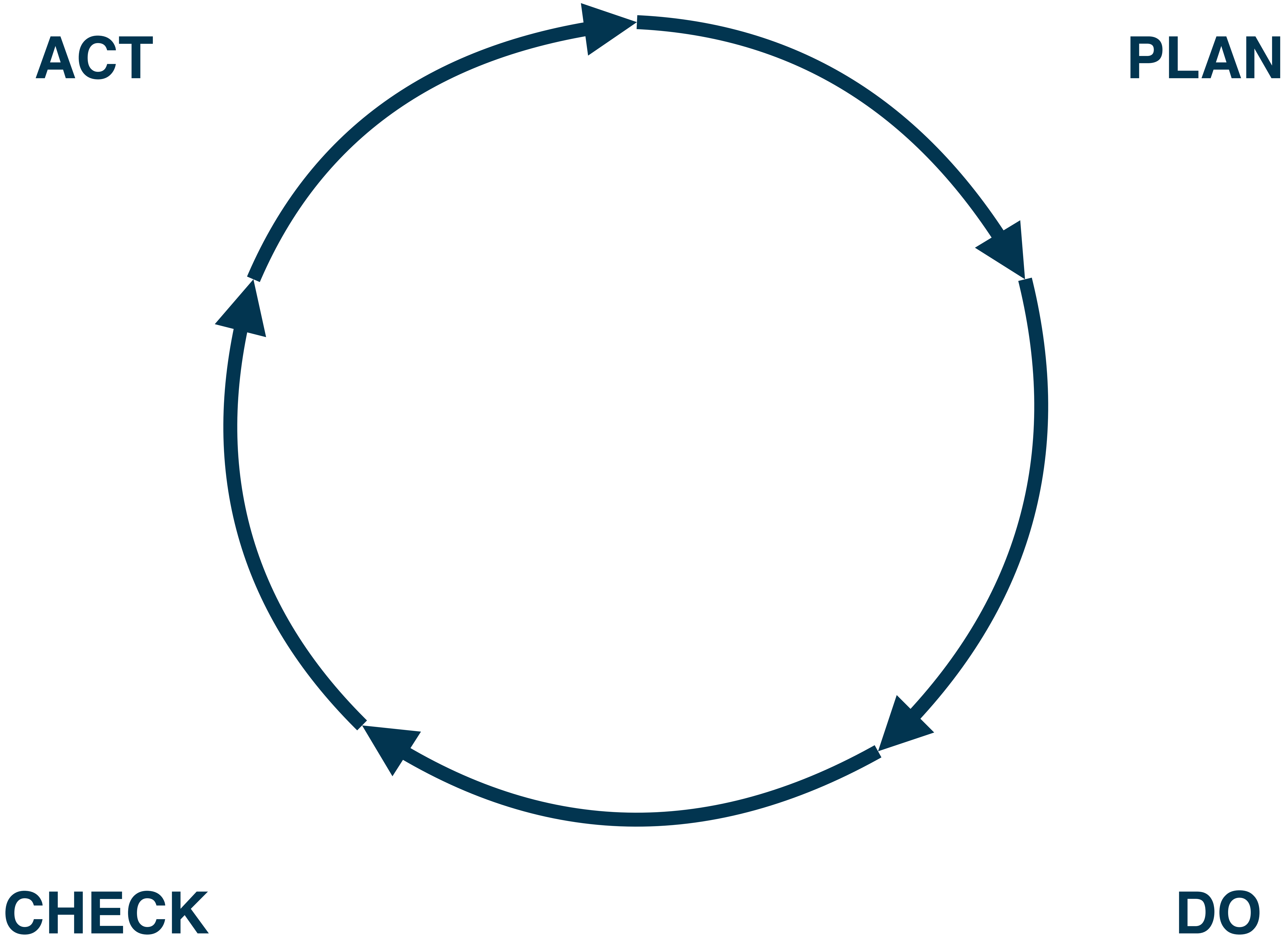






WHY MEASURE

FEEDBACK AND IMPROVEMENT



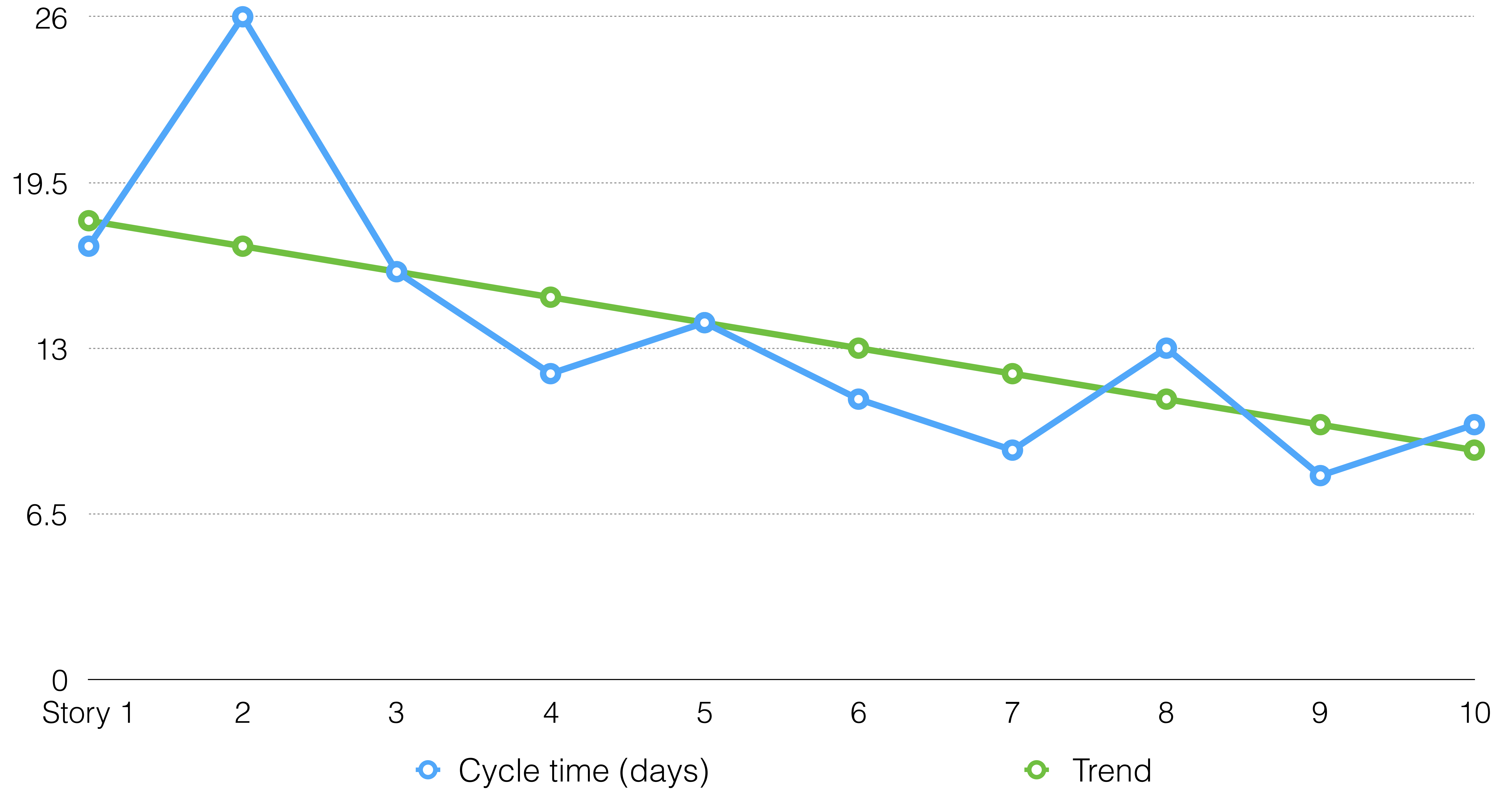
ACT

PLAN

CHECK

DO

PREDICTABILITY



BENCHMARKING

	High performers	Median performers	Low performers
Deployment frequency	On demand (multiple deploys per day)	1/week - 1/month	1/week - 1/month
Lead time for changes	<1hr	1 week - 1 month	1 week - 1 month
Change failure rate	<15%	<15%	31-45%
MTTR	<1hr	<1 day	1 day - 1 week

Modified from Forsgren PhD, Nicole. Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations

WHAT TO MEASURE

WHAT TO MEASURE

- Throughput
- Deployment frequency
- Cycle time
- Lead time
- Mean time between failures
- Mean time to recover (MTTR)

- Failure rate
- Defect fix times
- Escaped defects
- Total regression test time
- Number of branches in version control
- Production outages during deployment

WHAT TO MEASURE

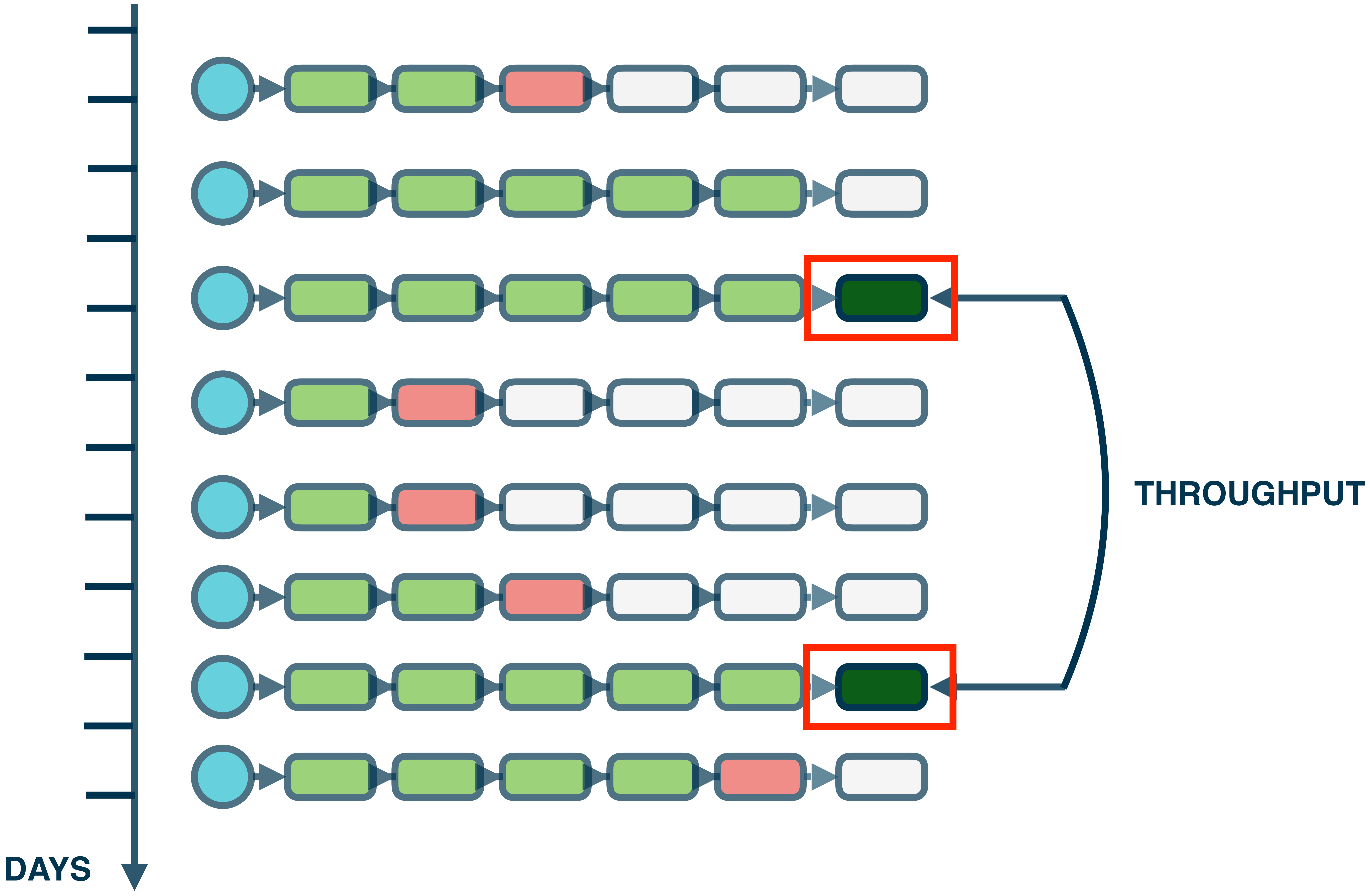
- **Throughput**
- Deployment frequency
- **Cycle time**
- Lead time
- Mean time between failures
- **Mean time to recover (MTTR)**

- **Failure rate**
- Defect fix times
- Escaped defects
- Total regression test time
- Number of branches in version control
- Production outages during deployment

THROUGHPUT

How often does code reach a certain point in the CD pipeline?

E.g. How often do you deploy?

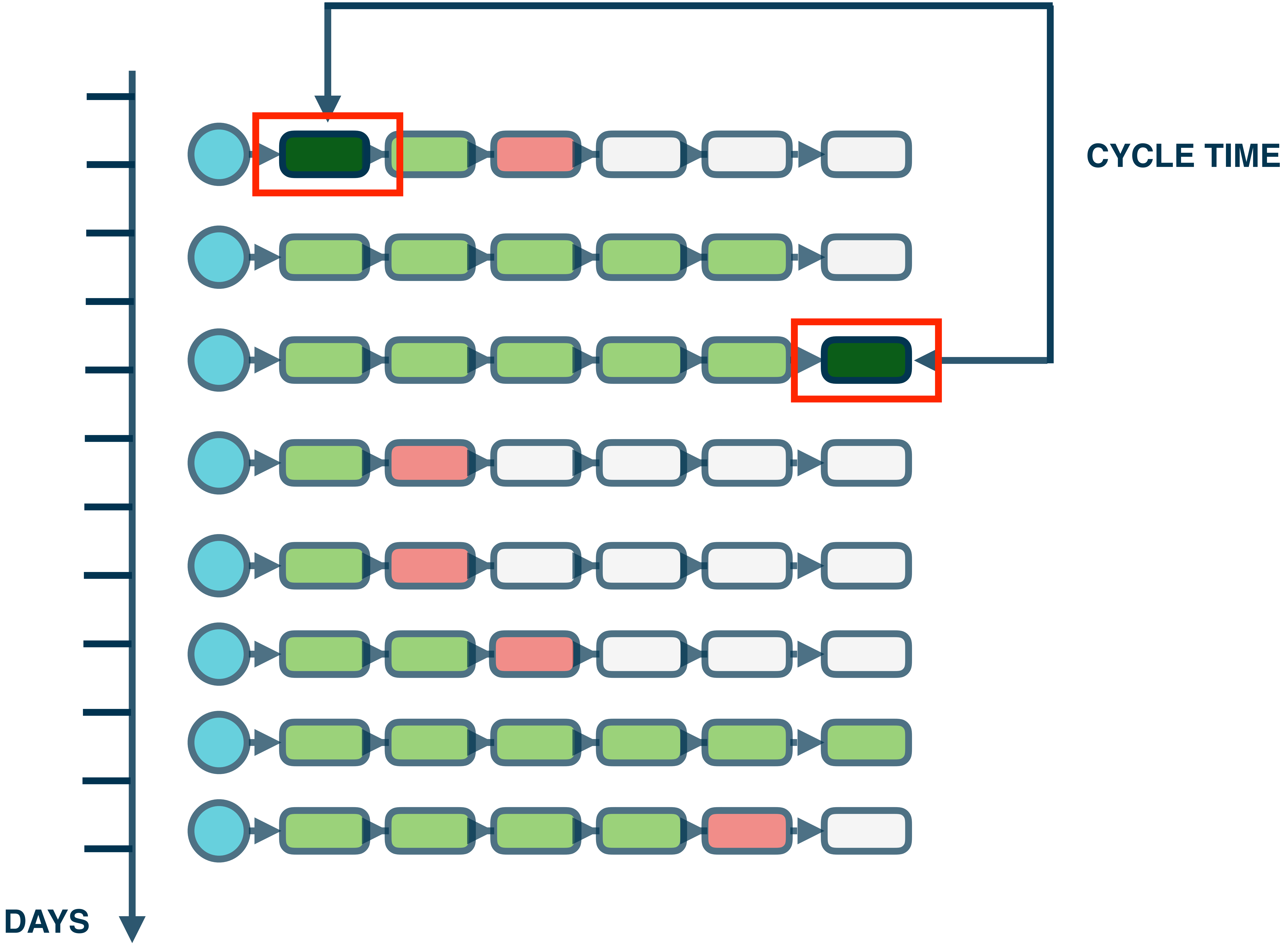


THROUGHPUT: 25%

CYCLE TIME

How long does it take to go from one point to the the to another point in the CD pipeline?

E.g. How long does it take to go from code commit to code successfully running in production?



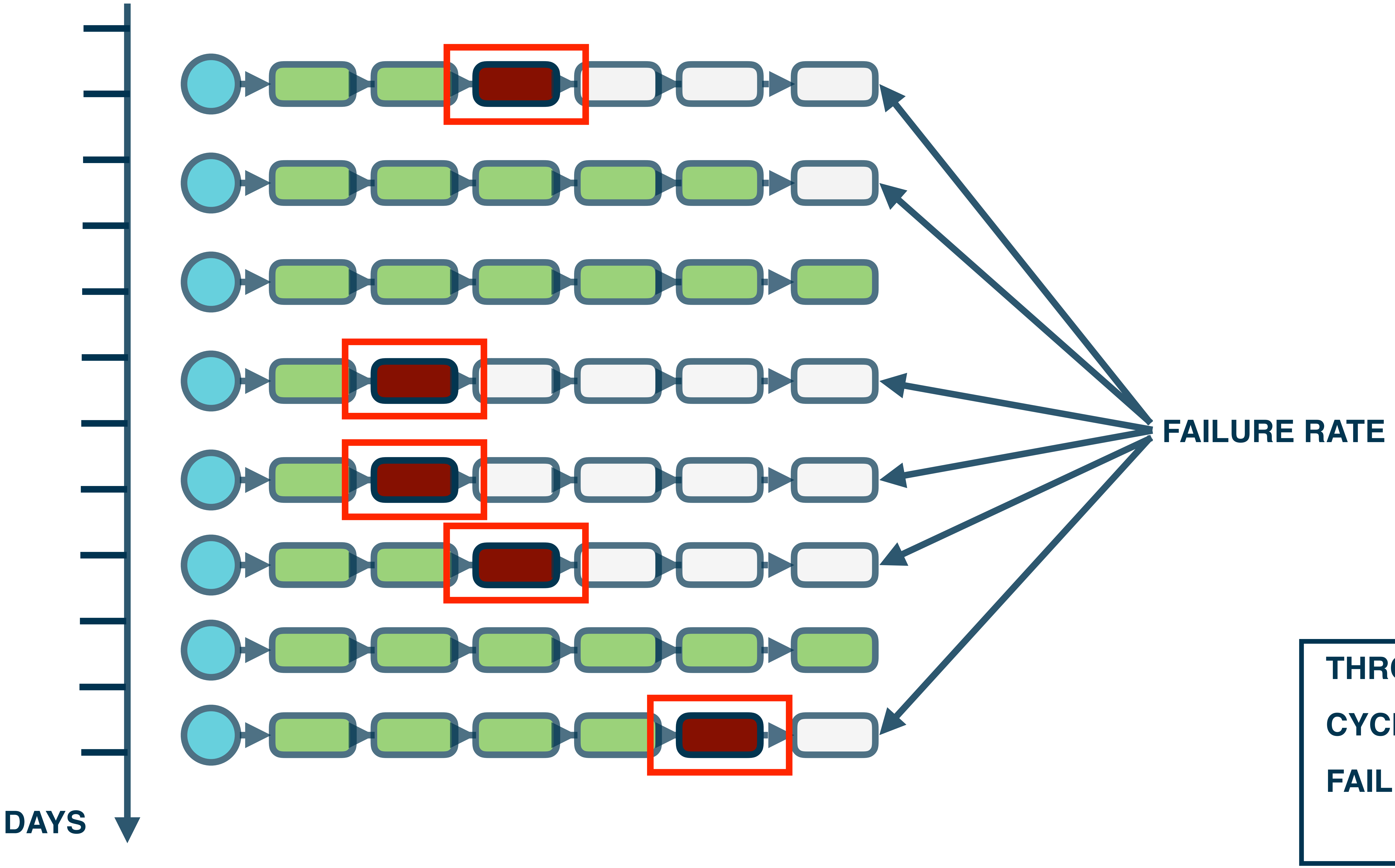
THROUGHPUT: 25%

CYCLE TIME: 3 DAYS

FAILURE RATE

What percentage of changes results a failure?

E.g. What percentage of changes break builds? What percentage of deployments result in a service outage?



THROUGHPUT: 25%

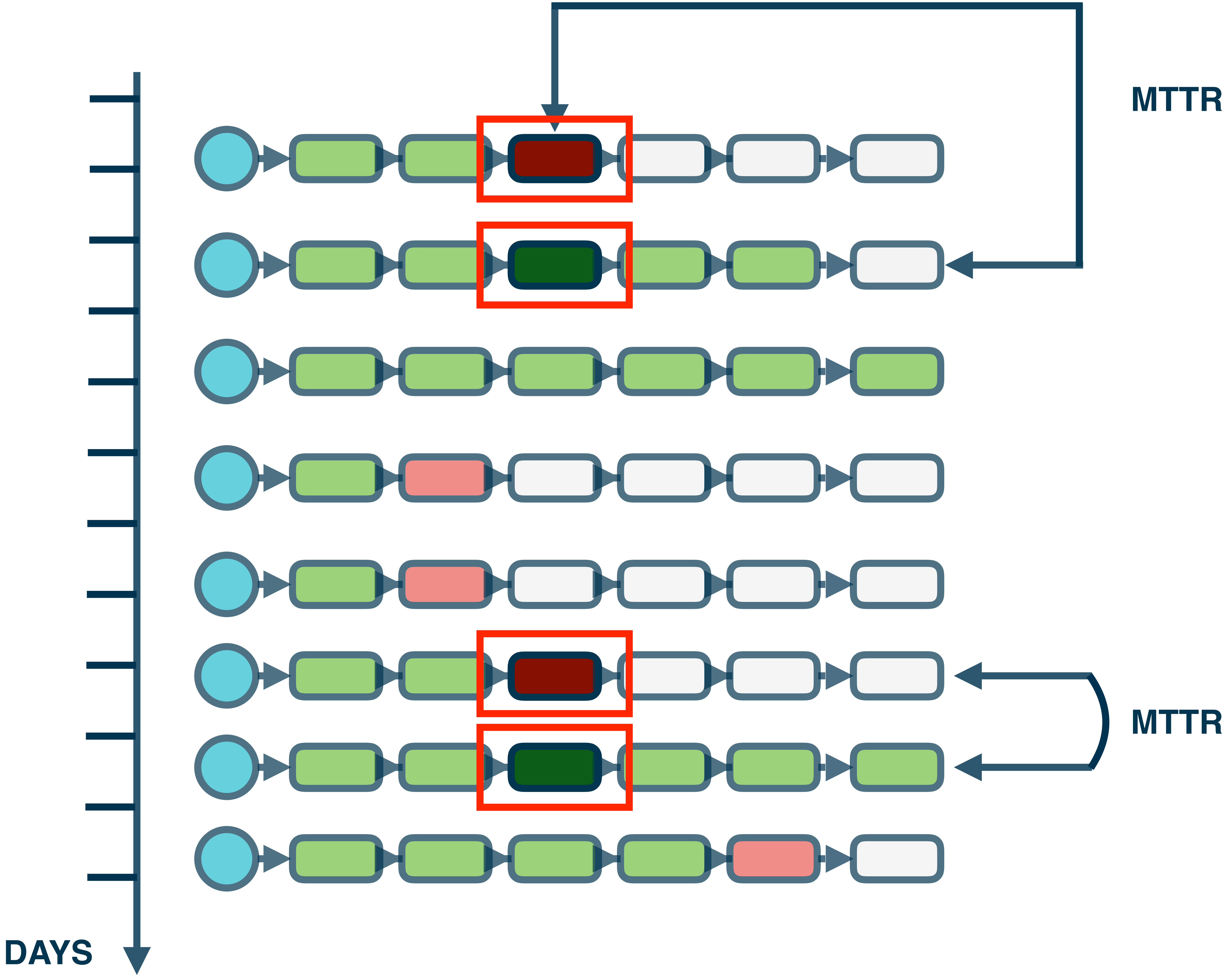
CYCLE TIME: 3 DAYS

FAILURE RATE: 75%

MEAN TIME TO RECOVER (MTTR)

How long does it generally take to fix a failure?

E.g. How long does it take to fix a broken build? How long does it take to restore service during a deployment failure?



THROUGHPUT: 25%

CYCLE TIME: 3 DAYS

FAILURE RATE: 75%

MTTR: 2 DAYS

WHAT TO MEASURE

- **Throughput**
- Deployment frequency
- **Cycle time**
- Lead time
- Mean time between failures
- **Mean time to recover (MTTR)**

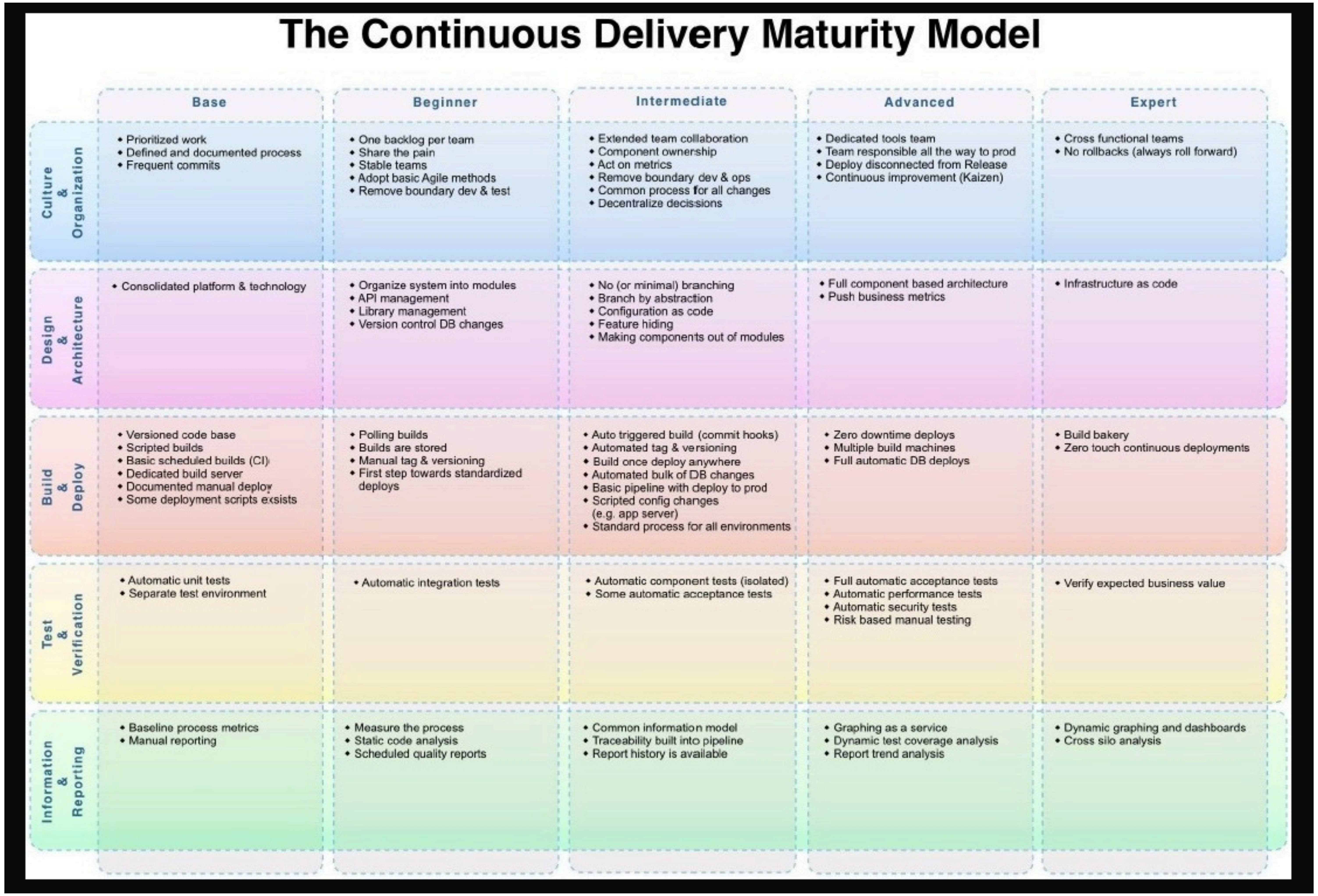
- **Failure rate**
- Defect fix times
- Escaped defects
- Total regression test time
- Number of branches in version control
- Production outages during deployment

CAUTION!

Beware of:

- vanity metrics
- unclear metrics
- invisible metrics
- comparing across teams
- the Hawthorne Effect or observer effect
- gathering “all the data” and not using it

VANITY METRICS



The Continuous Delivery Maturity Model

	Base	Beginner	Intermediate	Advanced	Expert
Culture & Organization	• Prioritized work • Defined and documented process • Frequent commits	• One backlog per team • Share the pain • Stable teams • Adopt basic Agile methods • Remove boundary dev & test	• Extended team collaboration • Component ownership • Act on metrics • Remove boundary dev & ops • Common process for all changes • Decentralize decissions	• Dedicated tools team • Team responsible all the way to prod • Deploy disconnected from Release • Continuous improvement (Kaizen)	• Cross functional teams • No rollbacks (always roll forward)
Design & Architecture	• Consolidated platform & technology	• Organize system into modules • API management • Library management • Version control DB changes	• No (or minimal) branching • Branch by abstraction • Configuration as code • Feature hiding • Making components out of modules	• Full component based architecture • Push business metrics	• Infrastructure as code
Build & Deploy	• Versioned code base • Scripted builds • Basic scheduled builds (CI) • Dedicated build server • Documented manual deploy • Some deployment scripts exsists	• Polling builds • Builds are stored • Manual tag & versioning • First step towards standardized deploys	• Auto triggered build (commit hooks) • Automated tag & versioning • Build once deploy anywhere • Automated bulk of DB changes • Basic pipeline with deploy to prod • Scripted config changes (e.g. app server) • Standard process for all environments	• Zero downtime deploys • Multiple build machines • Full automatic DB deploys	• Build bakery • Zero touch continuous deployments
Test & Verification	• Automatic unit tests • Separate test environment	• Automatic integration tests	• Automatic component tests (isolated) • Some automatic acceptance tests	• Full automatic acceptance tests • Automatic performance tests • Automatic security tests • Risk based manual testing	• Verify expected business value
Information & Reporting	• Baseline process metrics • Manual reporting	• Measure the process • Static code analysis • Scheduled quality reports	• Common information model • Traceability built into pipeline • Report history is available	• Graphing as a service • Dynamic test coverage analysis • Report trend analysis	• Dynamic graphing and dashboards • Cross silo analysis

https://res.infoq.com/articles/Continuous-Delivery-Maturity-Model/en/resources/fig1large.jpg

ACT ON YOUR METRICS

go
PIPELINES
ENVIRONMENTS
AGENTS
ANALYTICS
ADMIN
VSM
VSM
VSM
VSM
VSM

Value Stream Map

Pipeline
Production » Instance
4

git
/Users/aravindsv...
Initial commit
...

BuildAndUnitTests
Instance: 4
Duration: 25.0s

SmokeTests
Instance: 4
Duration: 35.0s

IntegrationTests
Instance: 4
Duration: 15.0s

Regression
Instance: 6
Duration: 15.0s

UAT
Instance: 6
Duration: 28.0s

Production
Instance: 4
Duration: In Progress

LOW THROUGHPUT

go
PIPELINES
ENVIRONMENTS
AGENTS
ANALYTICS
ADMIN

Value Stream Map

Pipeline
Production »
Instance
4

git
/Users/aravindsv...
Initial commit
...

BuildAndUnitTests
Instance: 4 VSM
Duration: 25.0s

SmokeTests
Instance: 4 VSM
Duration: 35.0s

IntegrationTests
Instance: 4 VSM
Duration: 15.0s

Regression
Instance: 6 VSM
Duration: 15.0s

UAT
Instance: 6 VSM
Duration: 28.0s

Production
Instance: 4
Duration: In Progress

go
PIPELINES
ENVIRONMENTS
AGENTS
ANALYTICS
ADMIN

Value Stream Map

Pipeline: Production » Instance: 4

git
/Users/aravindsv...
Initial commit
...

BuildAndUnitTests
Instance: 4 VSM
Duration: 25.0s

SmokeTests
Instance: 4 VSM
Duration: 35.0s

Regression
Instance: 6 VSM
Duration: 15.0s

UAT
Instance: 6 VSM
Duration: 28.0s

Production
Instance: 4
Duration: In Progress

Analytics for pipeline: BuildAndUnitTests

Pipeline Build Time

Last 24 Hours
Last 7 Days
Last 30 Days
All

Run Frequency	Mean Time to Recovery	Mean Time Between Failures	Failure Rate
103.00 per month	14h 5m	1d 18h 55m	41.67% (5 out of 12 runs)

Click and drag in the plot area to zoom in.
Shift + drag to pan horizontally.

go
PIPELINES
ENVIRONMENTS
AGENTS
ANALYTICS
ADMIN
git
/Users/aravindsv...
Initial commit
BuildAndUnitTests
SmokeTests
Regression
UAT
Production
VSM
Reset zoom
1h 20m 0.0s
1h 0m 0.0s
40m 0.0s
20m 0.0s
0.0s
Tue Jun 26
Wed Jul 4
Mon Jul 9
Fri Jul 20
Wait Time
Build Time
Mean Build Time

Value Stream Map

Pipeline: Production » Instance: 4

BuildAndUnitTests
Instance: 4
Duration: 25.0s

SmokeTests
Instance: 4
Duration: 35.0s

Regression
Instance: 6
Duration: 15.0s

UAT
Instance: 6
Duration: 28.0s

Production
Instance: 4
Duration: In Progress

Analytics for pipeline: BuildAndUnitTests

Pipeline Build Time

Last 24 Hours | Last 7 Days | Last 30 Days | All

Run Frequency
103.00 per month

Mean Time to Recovery
14h 5m

Mean Time Between Failures
1d 18h 55m

Failure Rate
41.67% (5 out of 12 runs)

Click and drag in the plot area to zoom in.
Shift + drag to pan horizontally.

go
PIPELINES
ENVIRONMENTS
AGENTS
ANALYTICS
ADMIN
git
/Users/aravindsv...
Initial commit
Last 24 Hours
Last 7 Days
Last 30 Days
All
Reset zoom
1h 20m 0.0s
1h 0m 0.0s
40m 0.0s
20m 0.0s
0.0s
Tue Jun 26
Wed Jul 4
Mon Jul 9
Fri Jul 20
40m 0.0s
30m 0.0s
20m 0.0s
10m 0.0s
0.0s
Tue Jun.Tue Jun 19
Wed Jun 20
Wed Jun 27
Thu Jul 12
Wed Jul 18
Wait Time
Build Time
Mean Build Time

Value Stream Map

Pipeline: Production » Instance: 4

BuildAndUnitTests — Instance: 4 — VSM — Duration: 25.0s

SmokeTests — Instance: 4 — VSM — Duration: 35.0s

Regression — Instance: 6 — VSM — Duration: 15.0s

UAT — Instance: 6 — VSM — Duration: 28.0s

Production — Instance: 4 — Duration: In Progress

Analytics for pipeline: BuildAndUnitTests

Pipeline Build Time

Run Frequency 103.00 per month

Mean Time to Recovery 14h 5m

Mean Time Between Failures 1d 18h 55m

Failure Rate 41.67% (5 out of 12 runs)

Click and drag in the plot area to zoom in.
Shift + drag to pan horizontally.

Analytics for pipeline: Production

Pipeline Build Time

Run Frequency 17.00 per month

Mean Time to Recovery 19h 12m 12.290s

Mean Time Between Failures 0.0s

Failure Rate 5.88% (1 out of 17 runs)

Click and drag in the plot area to zoom in.
Shift + drag to pan horizontally.

LOW THROUGHPUT

Causes

- Slow builds
- Builds that fail often
- Long lived feature branches

How to resolve

- Review cycle time and failure rates
- Consider using feature toggles and short-lived branches

SLOW CYCLE TIME

go
PIPELINES
ENVIRONMENTS
AGENTS
ANALYTICS
ADMIN
Analytics
git
VSM

Value Stream Map

Pipeline
Production » Instance 789

/Users/aravindsv...
Initial commit
...

BuildAndUnitTests
Instance: 4
Duration: 25.0s

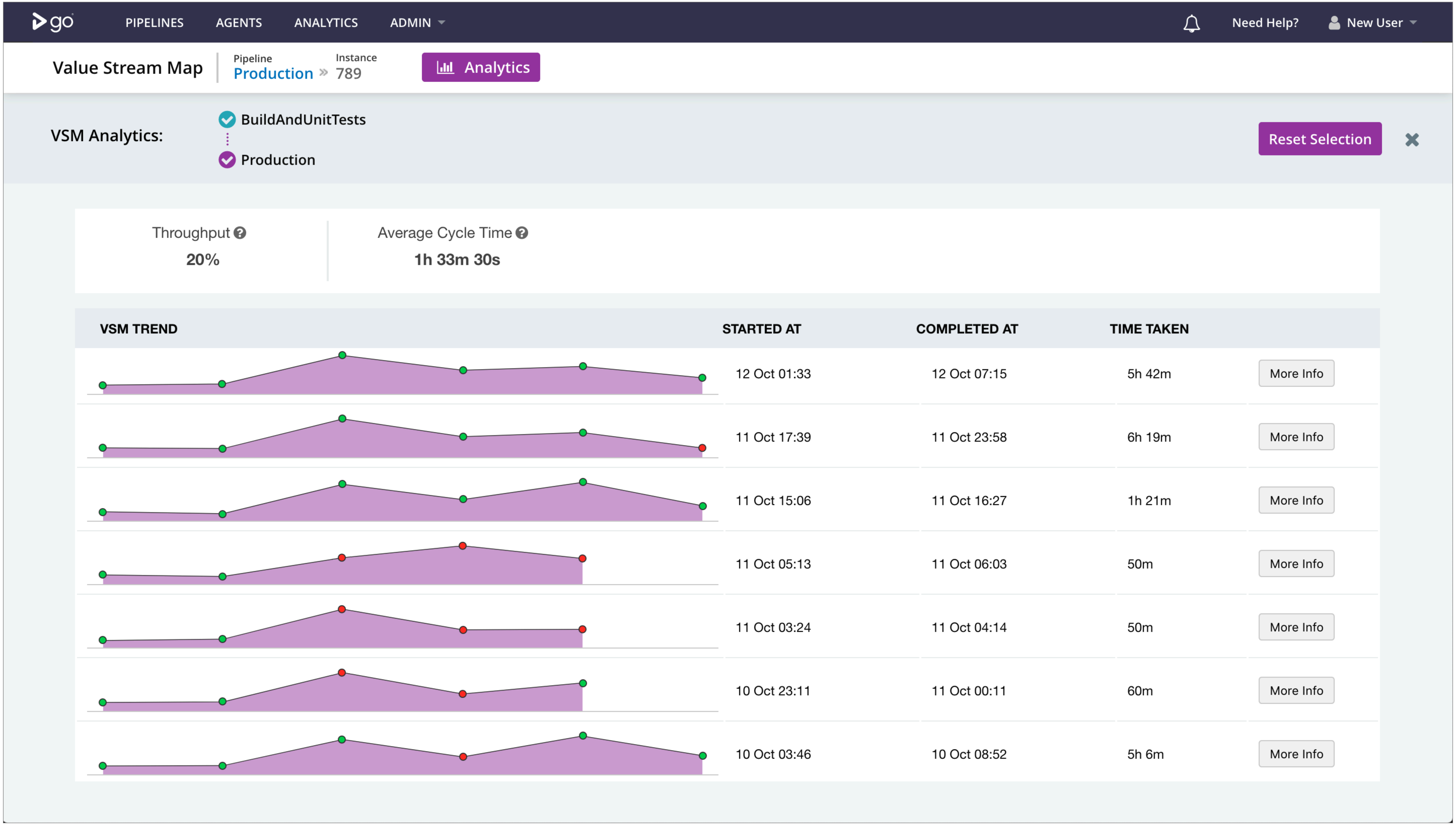
SmokeTests
Instance: 4
Duration: 35.0s

IntegrationTests
Instance: 4
Duration: 15.0s

Regression
Instance: 6
Duration: 15.0s

UAT
Instance: 6
Duration: 28.0s

Production
Instance: 4
Duration: In Progress



Value Stream Map

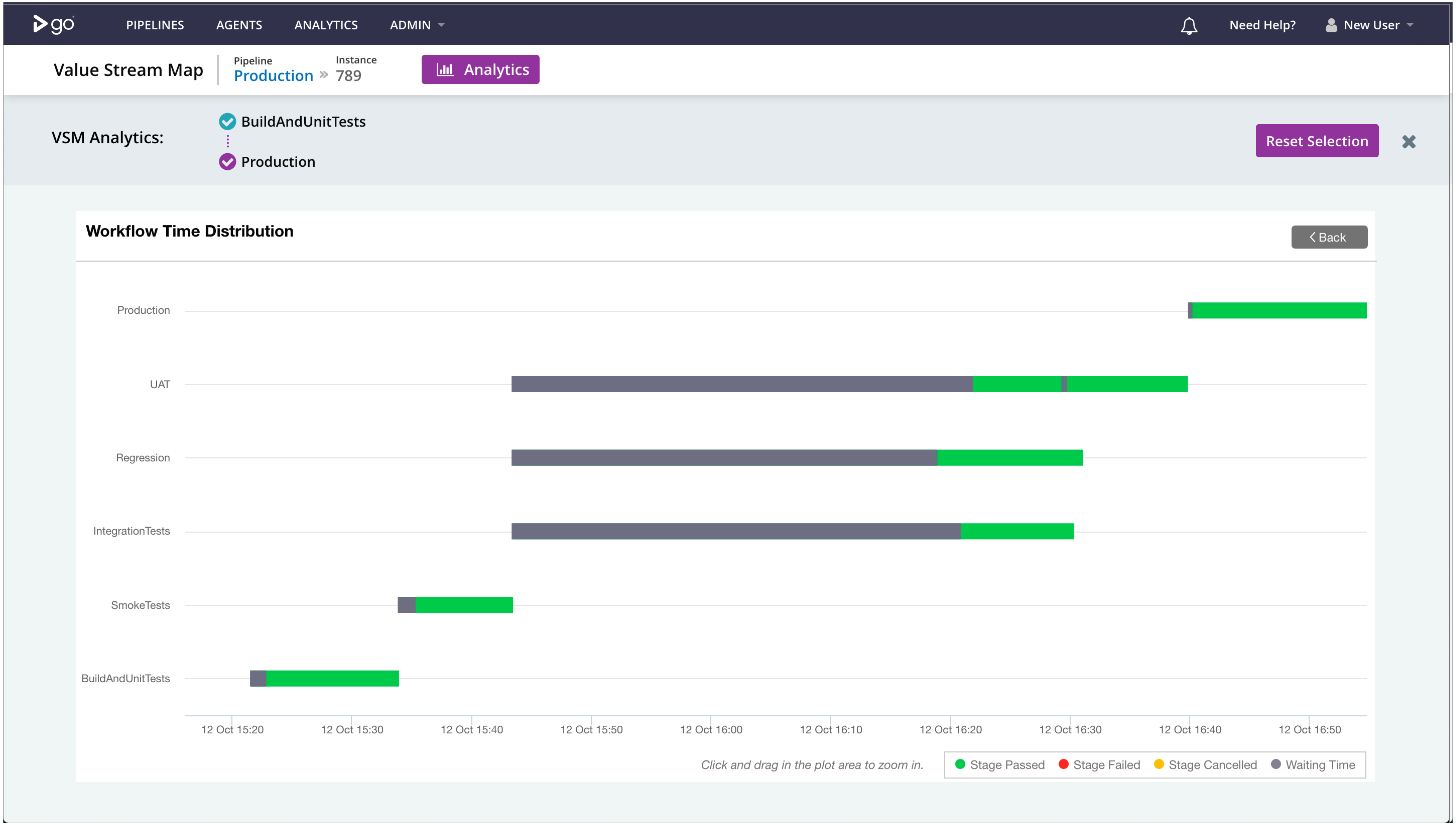
Pipeline: Production » Instance: 789

VSM Analytics: BuildAndUnitTests ⋮ Production

Throughput: 20%

Average Cycle Time: 1h 33m 30s

VSM TREND	STARTED AT	COMPLETED AT	TIME TAKEN	
	12 Oct 01:33	12 Oct 07:15	5h 42m	More Info
	11 Oct 17:39	11 Oct 23:58	6h 19m	More Info
	11 Oct 15:06	11 Oct 16:27	1h 21m	More Info
	11 Oct 05:13	11 Oct 06:03	50m	More Info
	11 Oct 03:24	11 Oct 04:14	50m	More Info
	10 Oct 23:11	11 Oct 00:11	60m	More Info
	10 Oct 03:46	10 Oct 08:52	5h 6m	More Info



Value Stream Map

Pipeline: Production » Instance: 789

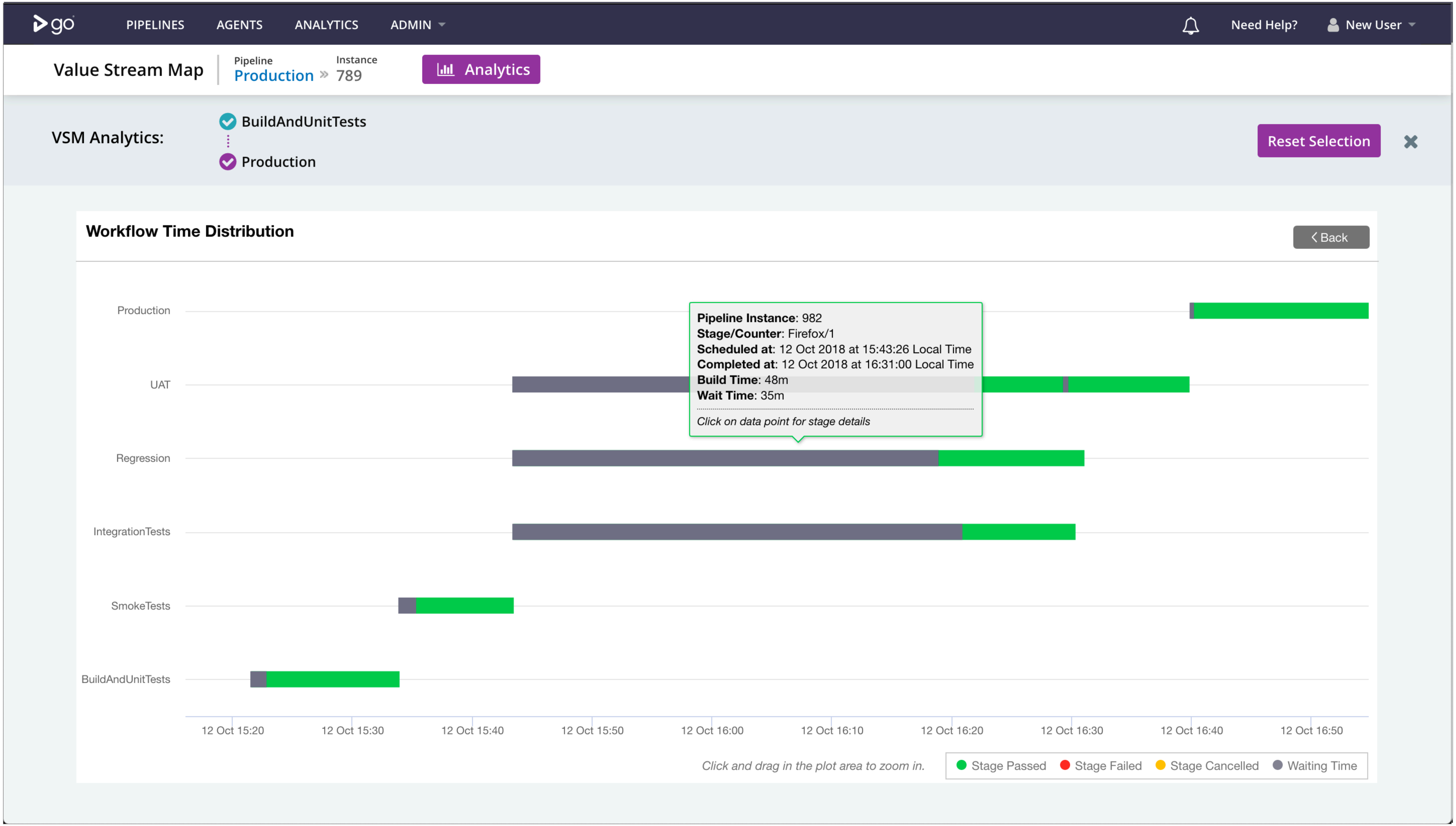
VSM Analytics:

✓ BuildAndUnitTests

✓ Production

Workflow Time Distribution

Click and drag in the plot area to zoom in.



Pipeline Instance: 982
Stage/Counter: Firefox/1
Scheduled at: 12 Oct 2018 at 15:43:26 Local Time
Completed at: 12 Oct 2018 at 16:31:00 Local Time
Build Time: 48m
Wait Time: 35m

Click on data point for stage details

SLOW CYCLE TIME

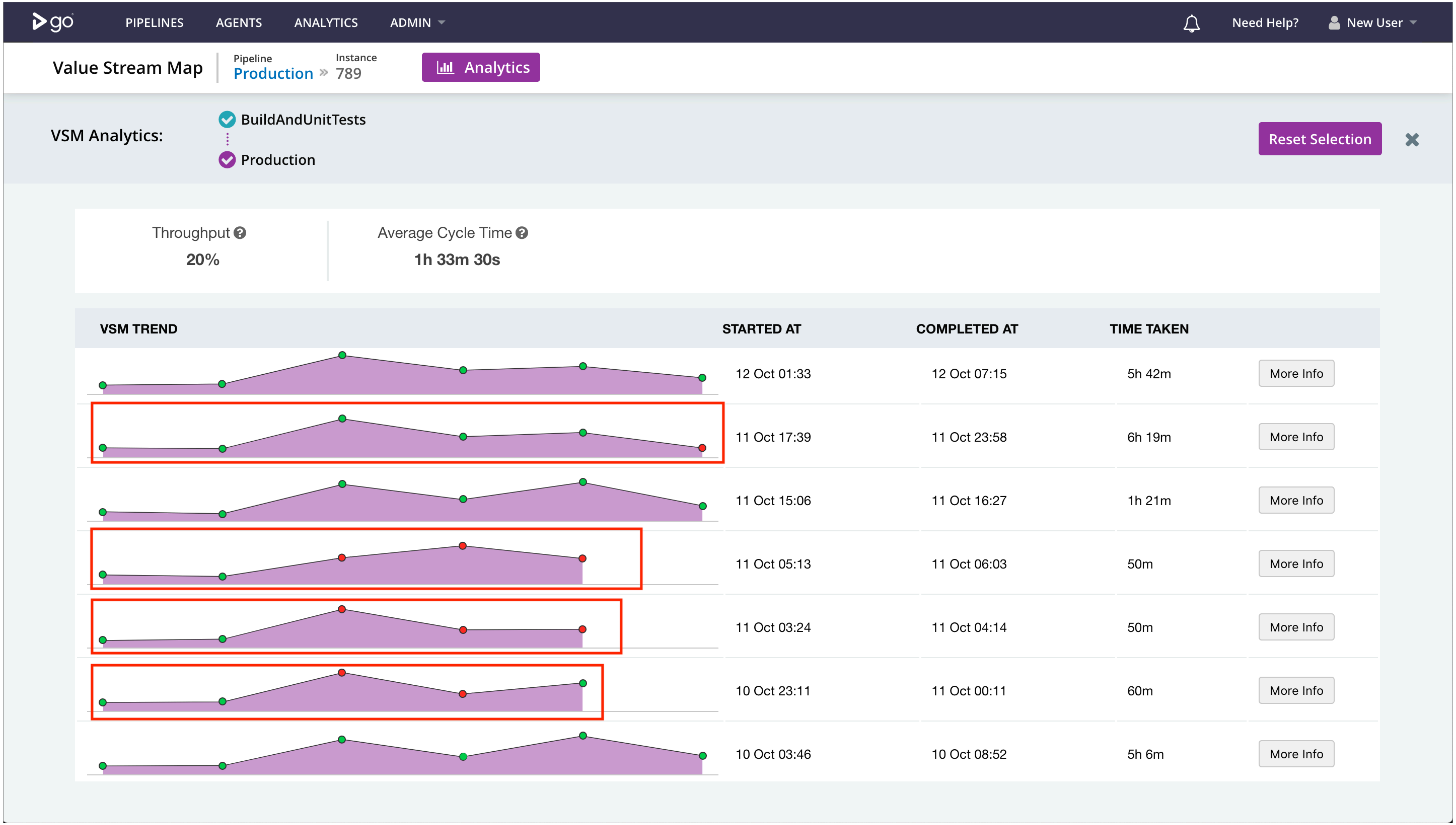
Causes:

- slow individual builds
- delays due to manual approvals

How to resolve:

- speed up slow steps by rewriting or parallelizing
- automate or simplify manual processes

HIGH FAILURE RATE



Value Stream Map

Pipeline Production » Instance 789

VSM Analytics:

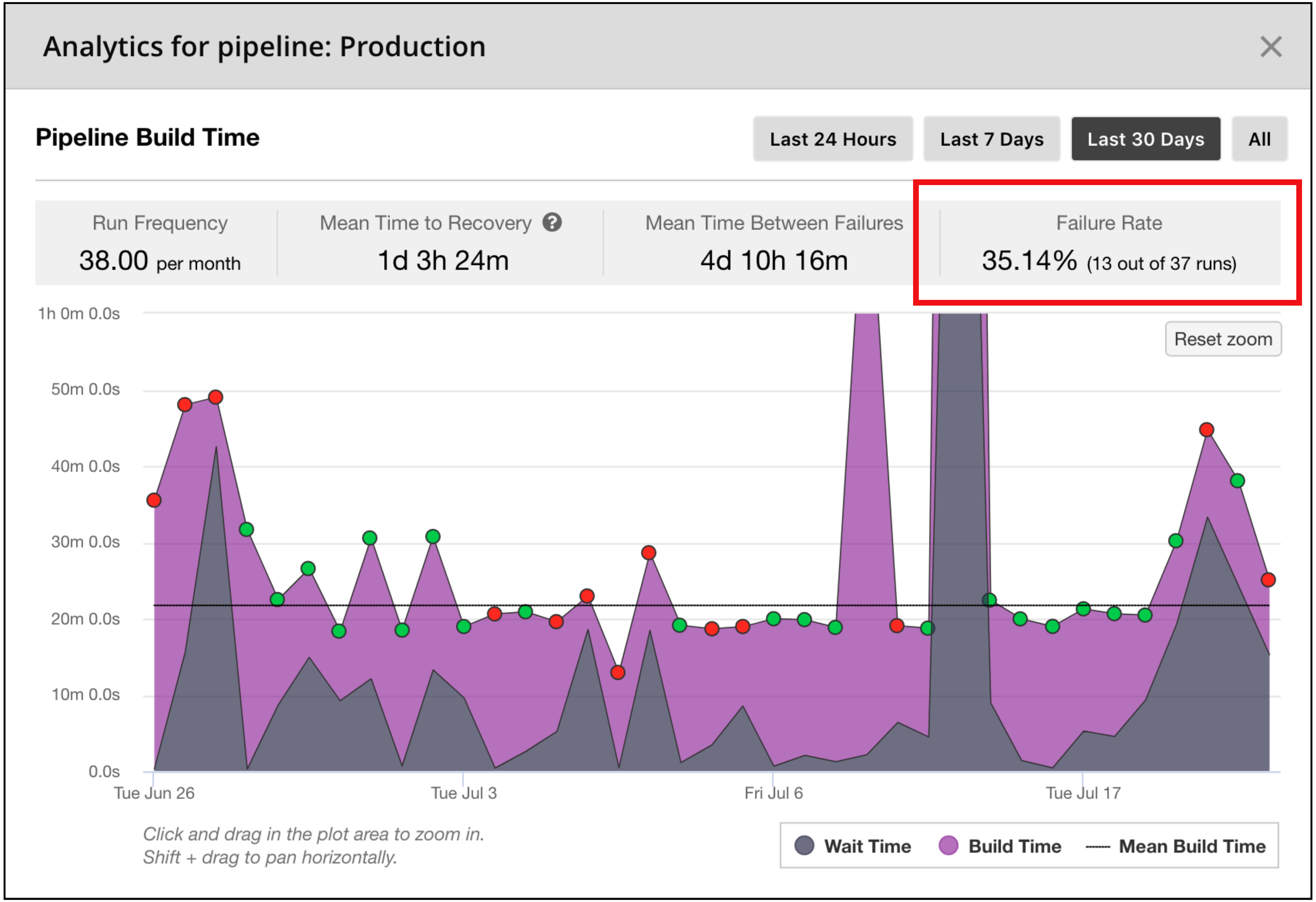
BuildAndUnitTests

Production

Throughput: 20%

Average Cycle Time: 1h 33m 30s

VSM TREND	STARTED AT	COMPLETED AT	TIME TAKEN	
	12 Oct 01:33	12 Oct 07:15	5h 42m	More Info
	11 Oct 17:39	11 Oct 23:58	6h 19m	More Info
	11 Oct 15:06	11 Oct 16:27	1h 21m	More Info
	11 Oct 05:13	11 Oct 06:03	50m	More Info
	11 Oct 03:24	11 Oct 04:14	50m	More Info
	10 Oct 23:11	11 Oct 00:11	60m	More Info
	10 Oct 03:46	10 Oct 08:52	5h 6m	More Info



Analytics for pipeline: Production

Pipeline Build Time

Run Frequency
38.00 per month

Mean Time to Recovery
1d 3h 24m

Mean Time Between Failures
4d 10h 16m

Failure Rate
35.14% (13 out of 37 runs)

Click and drag in the plot area to zoom in.
Shift + drag to pan horizontally.

HIGH FAILURE RATE

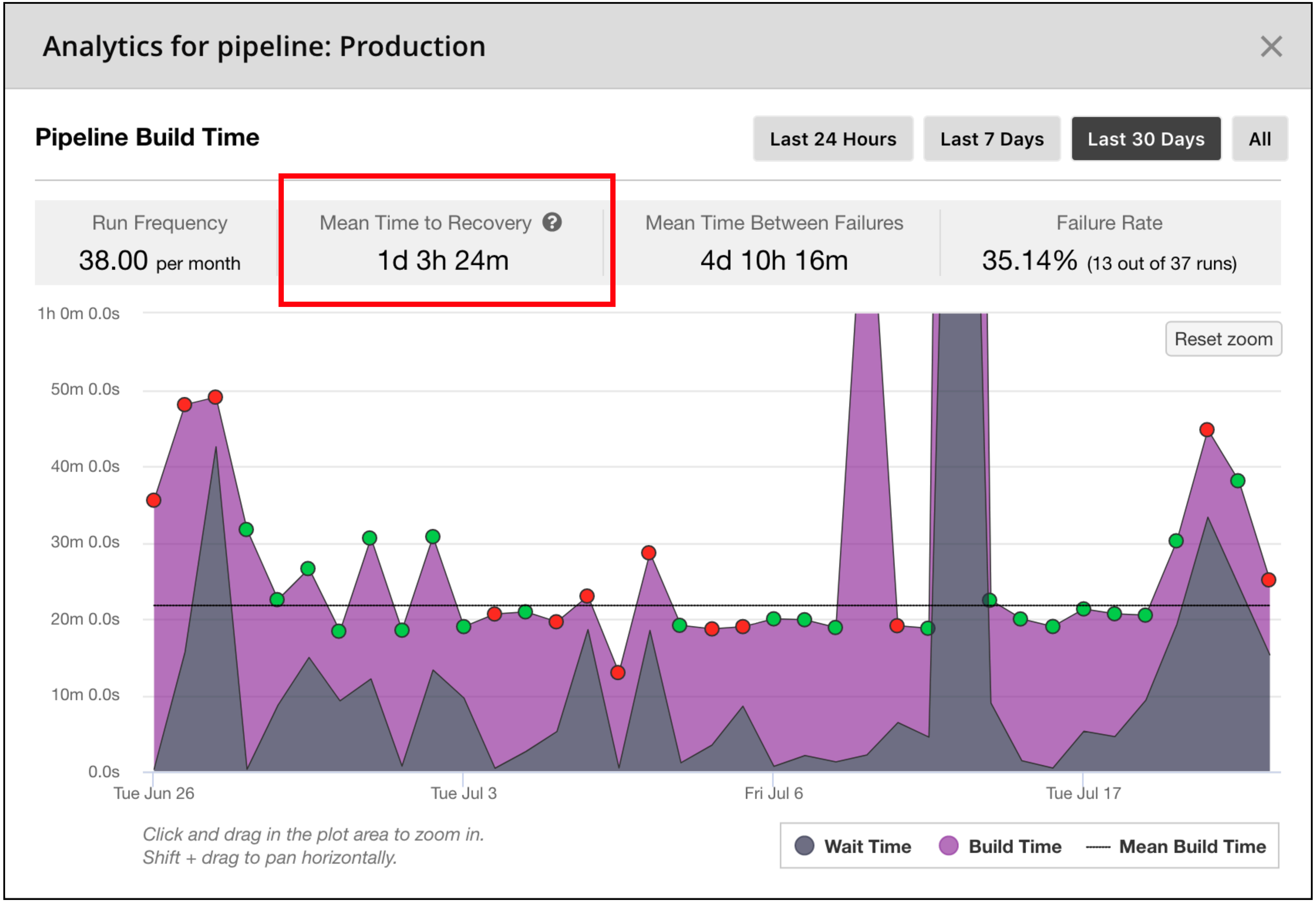
Causes

- genuine failures
- flaky tests
- tests too slow or difficult to run locally before check-in

How to resolve

- fail fast
- make it easier to run tests locally

HIGH MTTR



Analytics for pipeline: Production

Pipeline Build Time

Run Frequency
38.00 per month

Mean Time to Recovery ?
1d 3h 24m

Mean Time Between Failures
4d 10h 16m

Failure Rate
35.14% (13 out of 37 runs)

Click and drag in the plot area to zoom in.
Shift + drag to pan horizontally.

HIGH MTTR

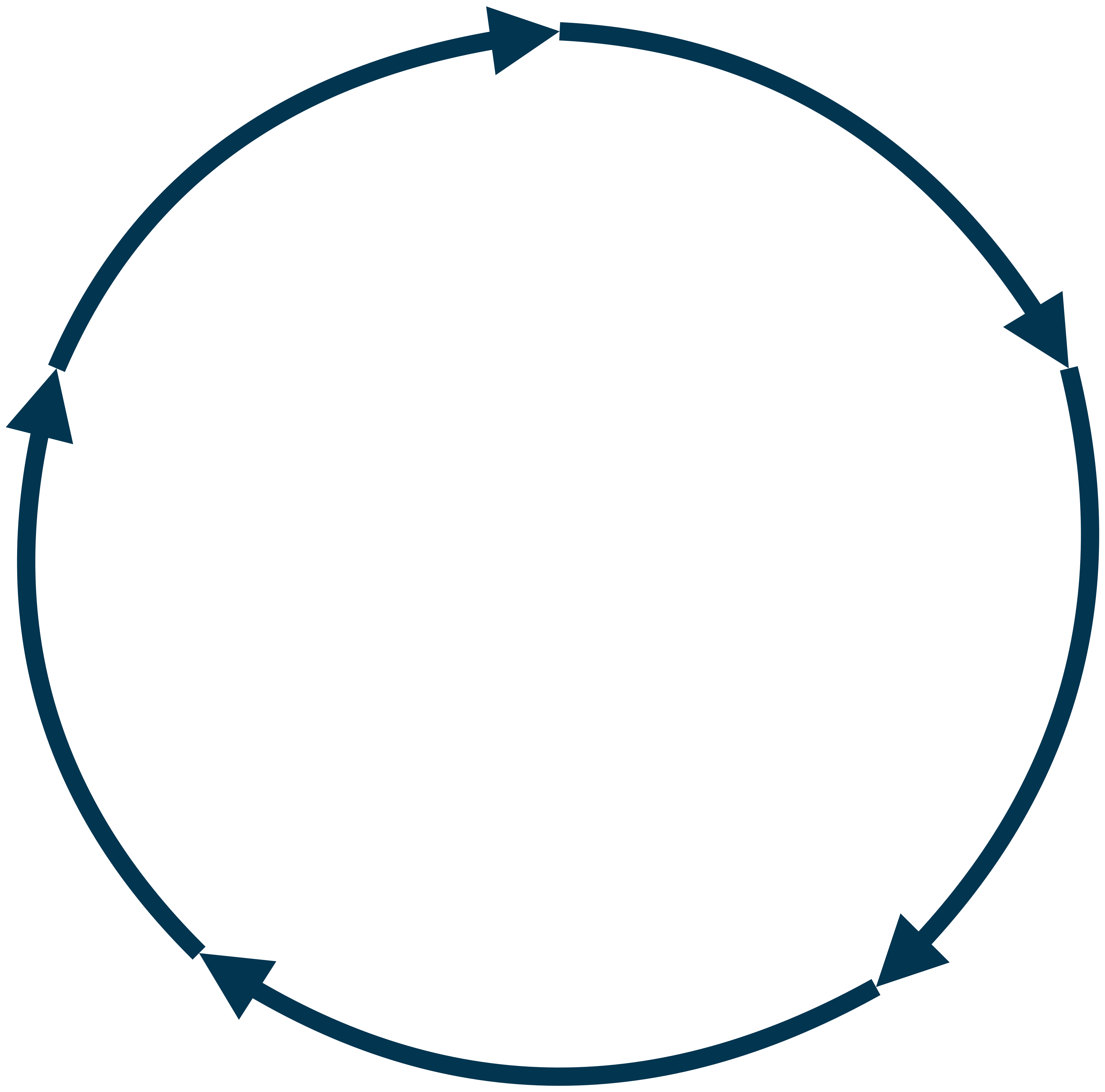
Causes

- no-one cares
- issues are hard to resolve
- combination with high failure rate and/or slow cycle time

How to resolve

- revert of failing commits
- stop the line

FEEDBACK CYCLES



ACT

PLAN

CHECK

DO

RECAP

RECAP

- Metrics are important to set goals, improve and predict
- Start with throughput, cycle time, failure rate and MTTR
- Be thoughtful about what you measure
- Look for connections between metrics
- Understand your context
- Review, change and improve your process
- Consider using tools to help capture and visualize data

ADDITIONAL RESOURCES

- Download GoCD - https://www.gocd.org/
- GoCD Analytics plugin - https://www.gocd.org/analytics/
- More events and talks - https://www.gocd.org/events/
- 4 important metrics for continuous delivery - https://www.gocd.org/2018/01/31/continuous-delivery-metrics/
- Why measure your CD process https://www.gocd.org/2018/10/30/measure-continuous-delivery-process/

QUESTIONS ?

ThoughtWorks®