Continuous Delivery

for the rest of us

About me

- Lisa van Gelder
- Consultant at Cyrus Innovation
- lvangelder@cyrusinnovation.com
- @techbint

Continuous Delivery

Is not:

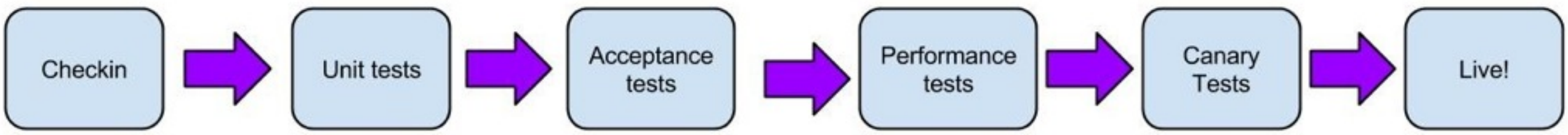
- Continuous deployment
- Automating all the things

Is:

- Removing the bottlenecks that stop you delivering

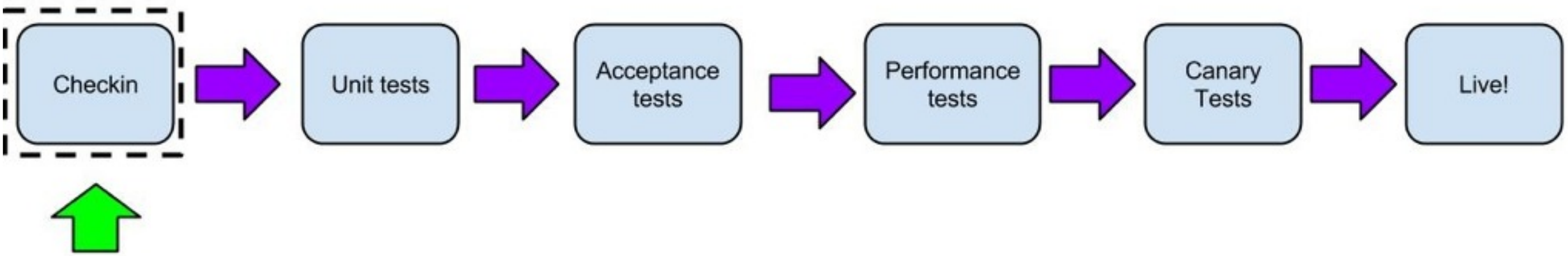
The Deployment Pipeline

Continuous Deployment



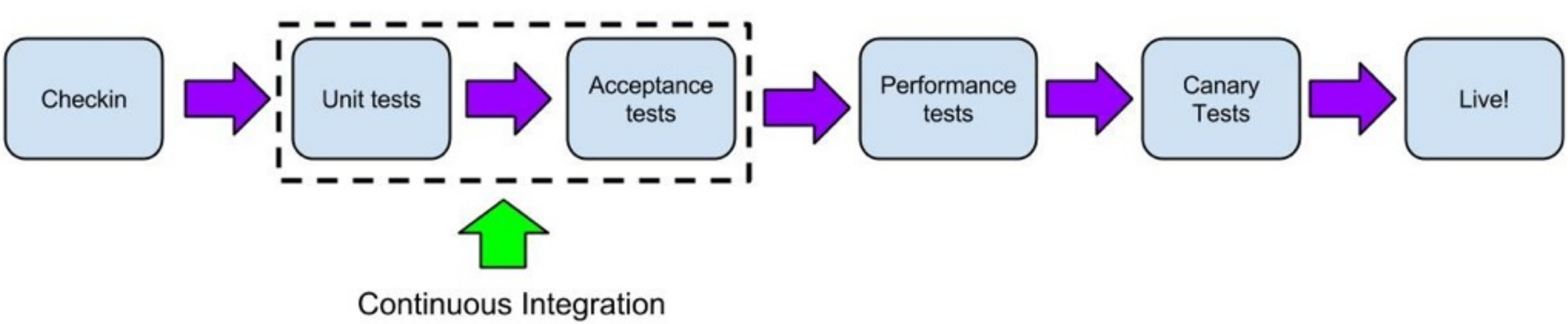
The Deployment Pipeline

Continuous Deployment



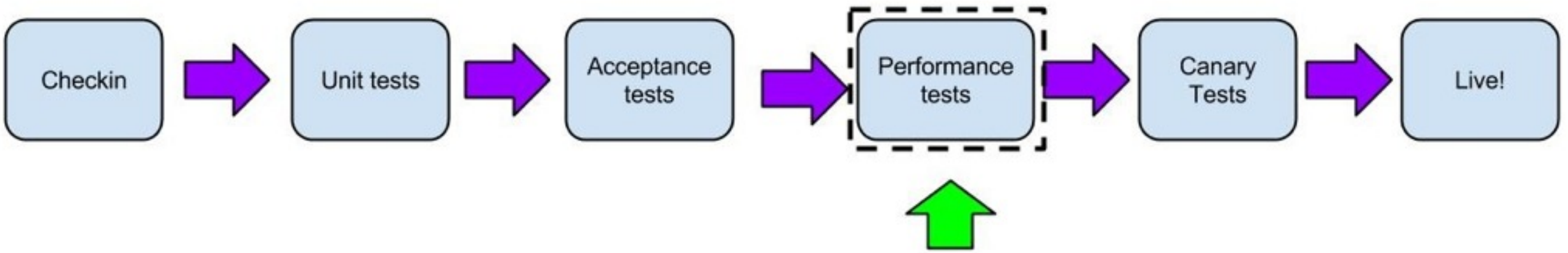
The Deployment Pipeline

Continuous Deployment



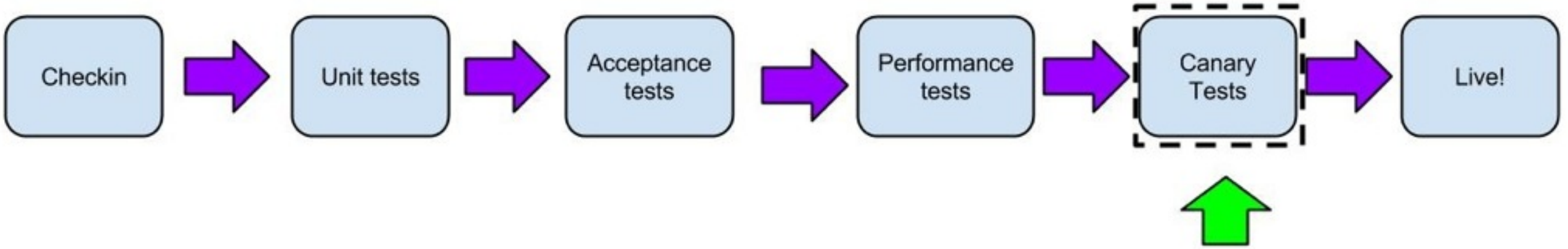
The Deployment Pipeline

Continuous Deployment



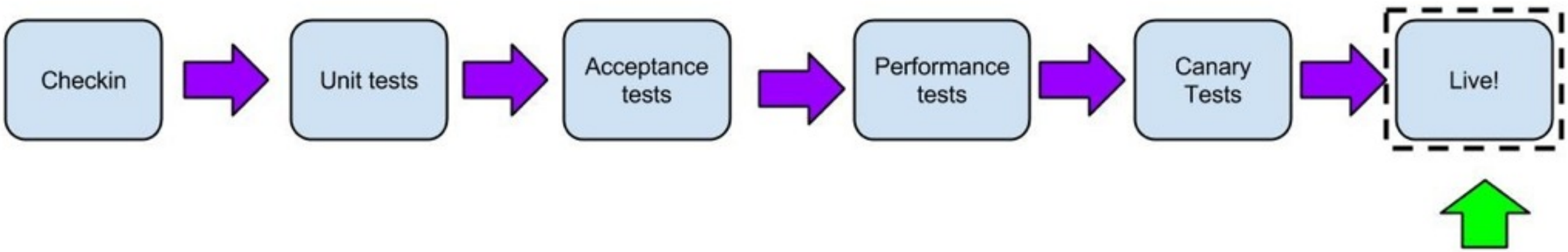
The Deployment Pipeline

Continuous Deployment



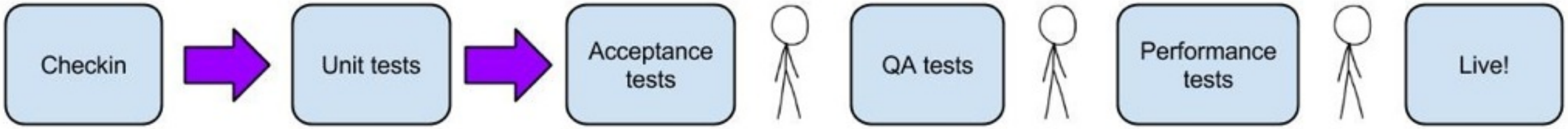
The Deployment Pipeline

Continuous Deployment



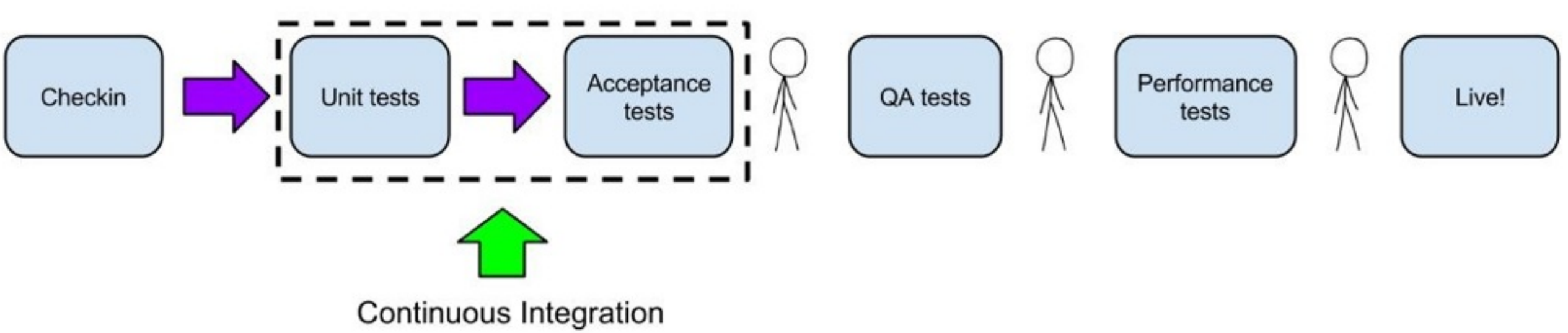
The Deployment Pipeline

Continuous Delivery



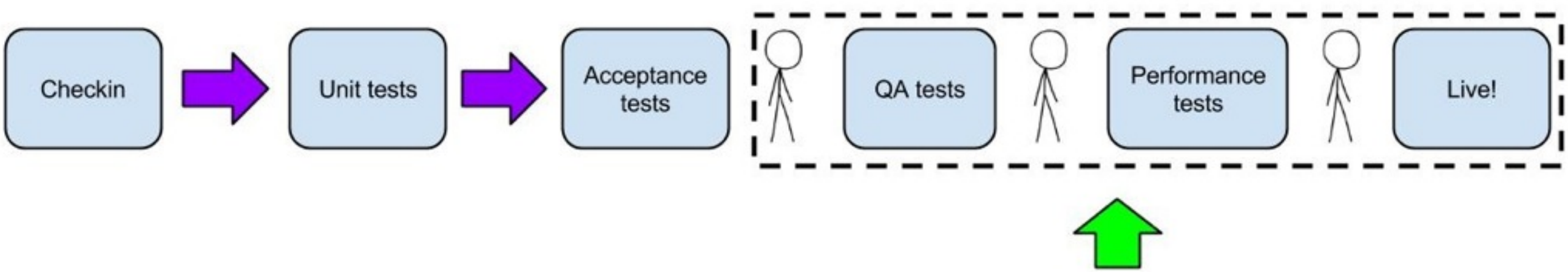
The Deployment Pipeline

Continuous Delivery



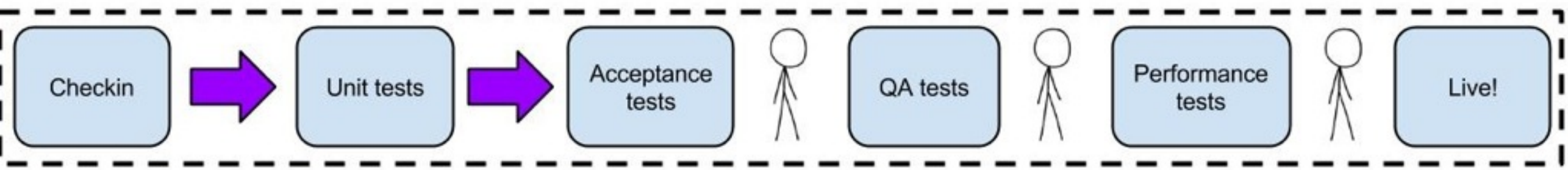
The Deployment Pipeline

Continuous Delivery



The Deployment Pipeline

Cycle Time



The time it takes from checkin to master to code being live in production

HL
12TH AF
RESCUE
0436

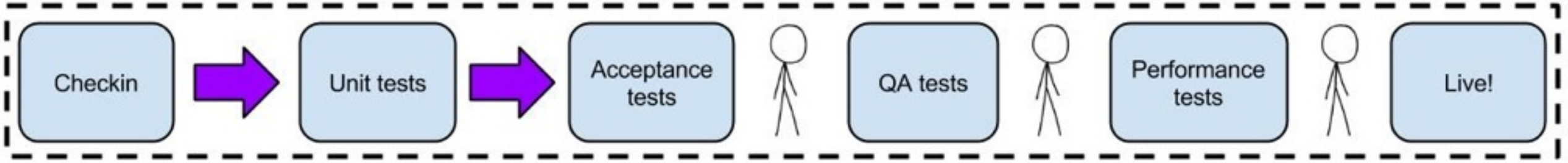
Ooda loop

- observe, orient, decide, and act

- John Boyd, Father of the F16

Cycle time

The time it takes from checkin to master to code being live in production



Cycle Time is the act phase of your ooda loop

Cycle time is the reaction time of your organization

Continuous Delivery

- Measure the total cycle time
- Reduce your cycle time
- Improve your reaction time

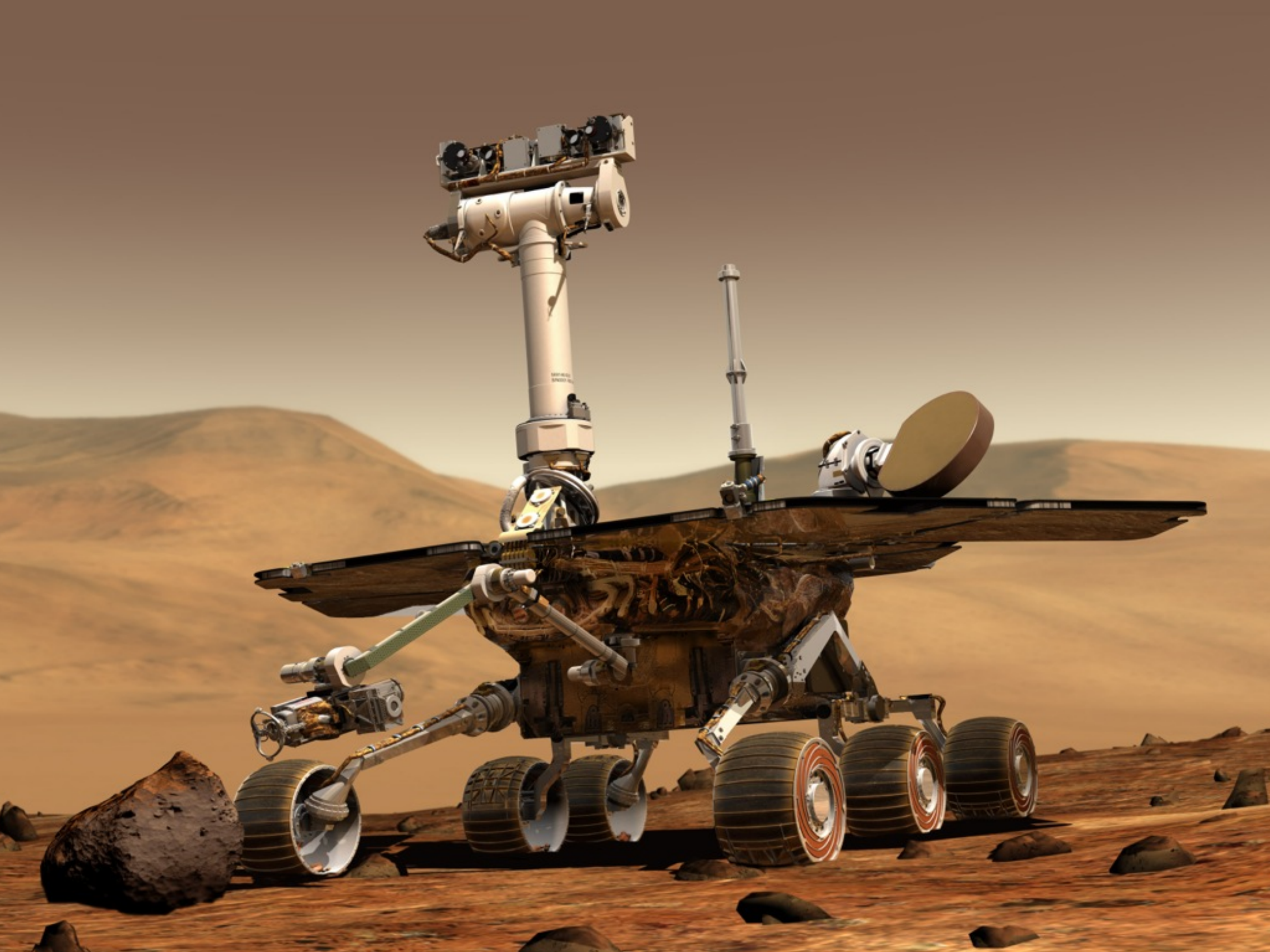
How do you choose where to start?

Double the frequency of releases

Benefits of smaller releases

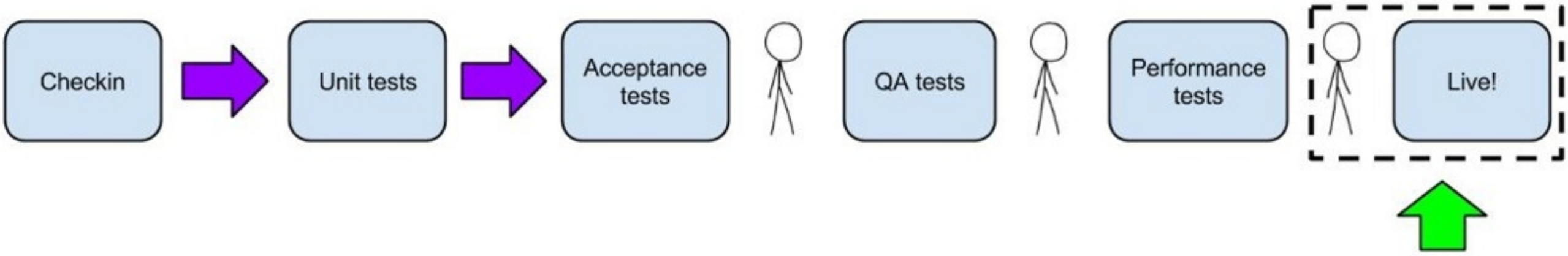
- Code only has value in production - get it there quicker!
- Less co-ordination required
- Easier to test
- Easier to see if it caused issues in production
- Easier to rollback

If you can’t release,
fake it



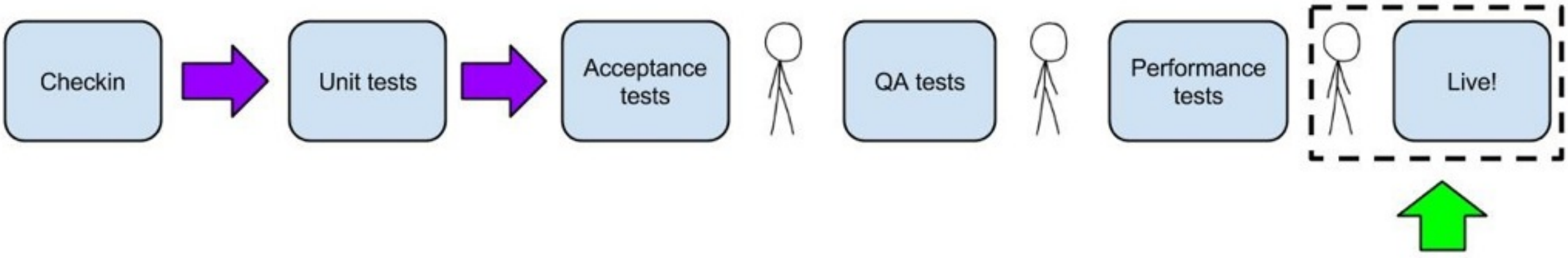
Blockers

- Releases cause problems for users
- Ops team don’t have time to do more releases
- Ops/dev don’t have time to support more releases
- QA don’t have time to test new features
- Takes too long to get a green build



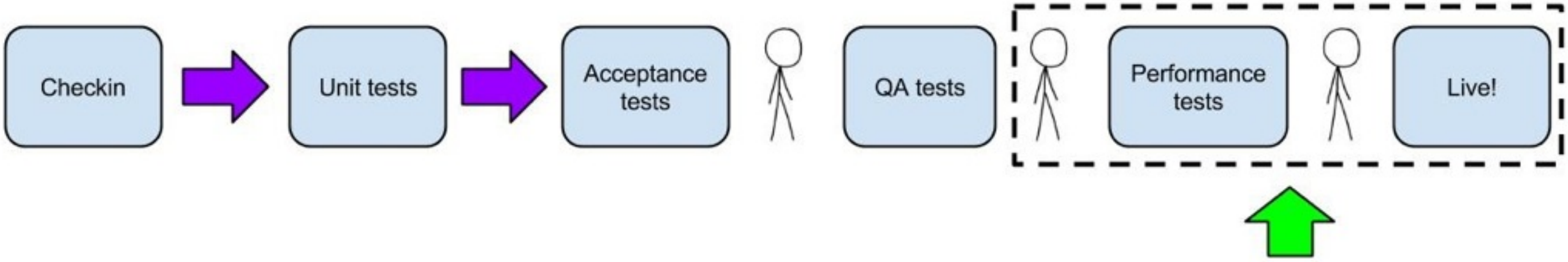
Releases cause problems for users

- data loss
- performance issues
- broken functionality



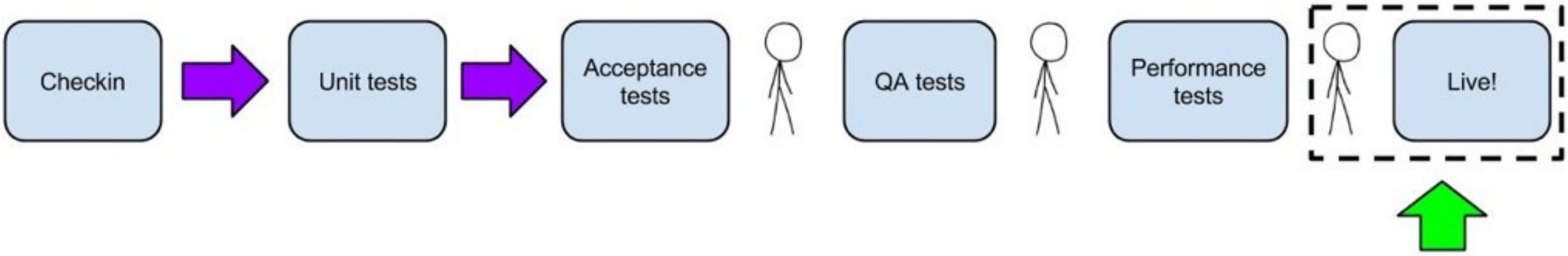
The cms that made journalists stop work for 20 minutes…

- db changes tied to code changes
- too much state in session



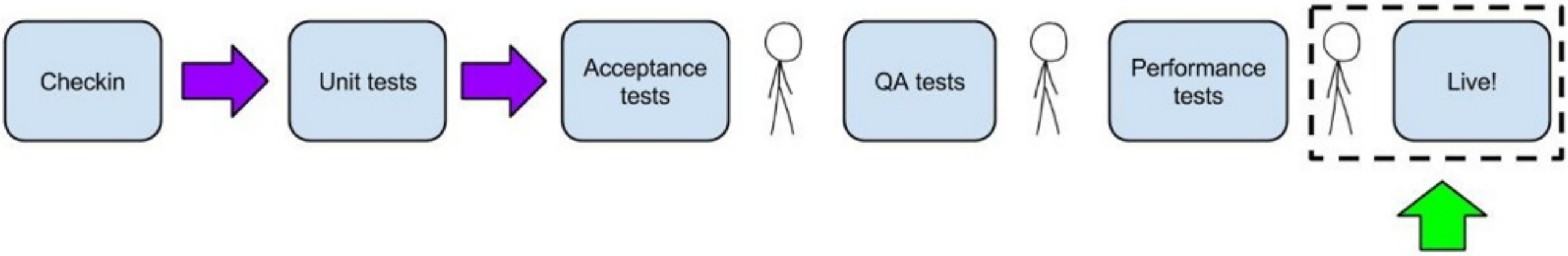
Releases cause performance problems

- play back logs
- soak test
- dark launch
- performance test as soon as you can



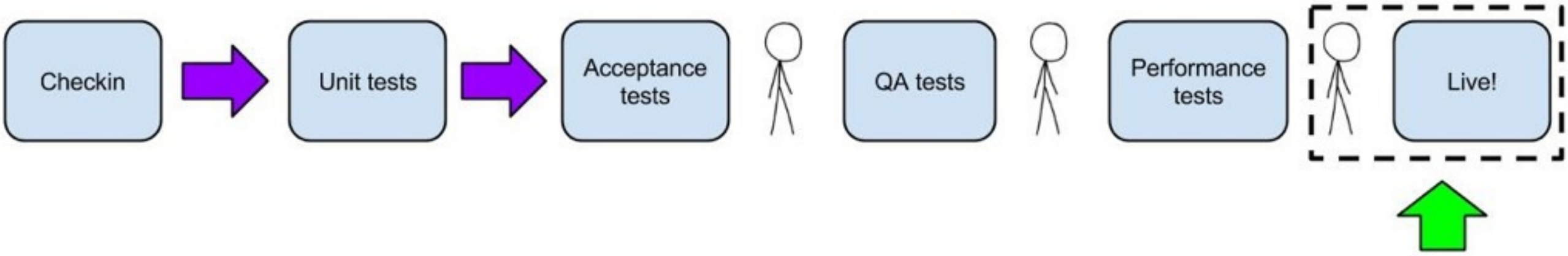
Releases break existing functionality

- missing tests
- no env like prod to test on



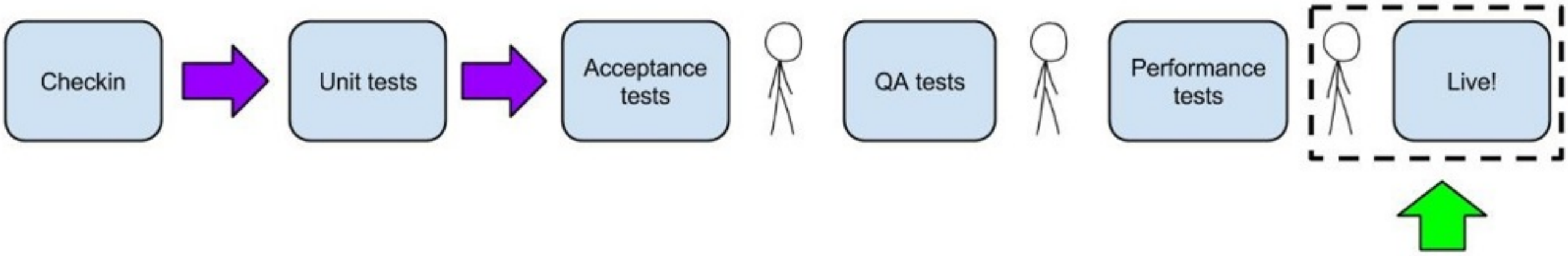
The cdn that broke the release

- POST /myapp/comment/123/recommend



Ops don’t have the time to perform more releases

ZS-NVH
ZS-NVH
FIRE DETECT
ANNUN PANEL
LAMP
NIGHT
TEST
TEST
DAY
MAX WT MANEUVER SPEED 148 KIAS
SEE POH FOR OTHER WEIGHTS
ALT
VERTICAL SPEED
VENT PULL ON
GEN
ALT
BATT
VOLT
LIGHTS
LEFT LDG
TAXI/ RECOG
RIGHT LDG
ON
STROBE
NAV
BCN
ON
ON
SEAT BELT
CABIN
ON
PITOT STATIC HEAT
STALL HEAT
ON
DEICE/ANTI-ICE WING LIGHT
W/S
PROP
AUTO
OFF
OFF
MANUAL
ALT STATIC AIR PULL ON
PARKING BRAKE PULL
LOCK
DISCONNECT AUTOPILOT AT FIRST INDICATION OF ICE ACCRETION
DO NOT TAKEOFF WITH ICE/FROST/SNOW ON AIRCRAFT.



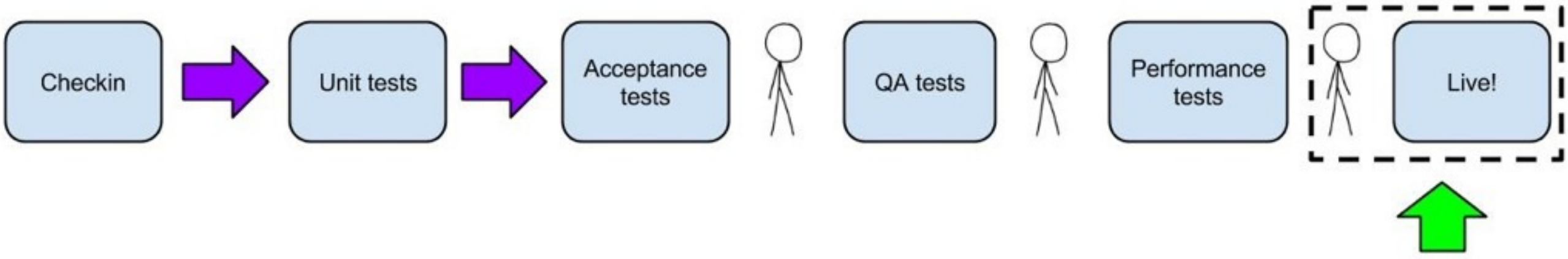
Ops/dev don’t have time to watch releases

- Release interrupts normal work
- Team on standby in case of issues

e>load c3
view c3
e>draw PROD.streambase.tbs-shop.all.jniIn.
PROD.streambase.tbs-shop.all.jniIn.LocationFinderService.
PROD.streambase.tbs-shop.all.jniIn.SessionEventService.
PROD.streambase.tbs-shop.all.jniIn.ShopService.
PROD.streambase.tbs-shop.all.jniIn.stdevLatency
PROD.streambase.tbs-shop.all.jniIn.avgLatency
PROD.streambase.tbs-shop.all.jniIn.count
PROD.streambase.tbs-shop.all.jniIn.countFailed
PROD.streambase.tbs-shop.all.jniIn.maxLatency
PROD.streambase.tbs-shop.all.jniIn.minLatency
commits
carbon.agents.grph01vip.committedPoints
carbon.agents.grph02vip.committedPoints
consumption
caches
carbon.agents.grph01vip.cache.queues
carbon.agents.grph02vip.cache.queues
carbon.agents.grph01vip.cache.size
carbon.agents.grph02vip.cache.size
updateLatency
carbon.agents.grph01vip.avgUpdateTime
carbon.agents.grph02vip.avgUpdateTime
cacheQueries
carbon.agents.grph01vip.cache.queries
carbon.agents.grph02vip.cache.queries
receives
updates
carbon.agents.grph01vip.updateOperations
carbon.agents.grph02vip.updateOperations
pointsPerUpdate
carbon.agents.grph01vip.pointsPerUpdate
carbon.agents.grph02vip.pointsPerUpdate

Help!

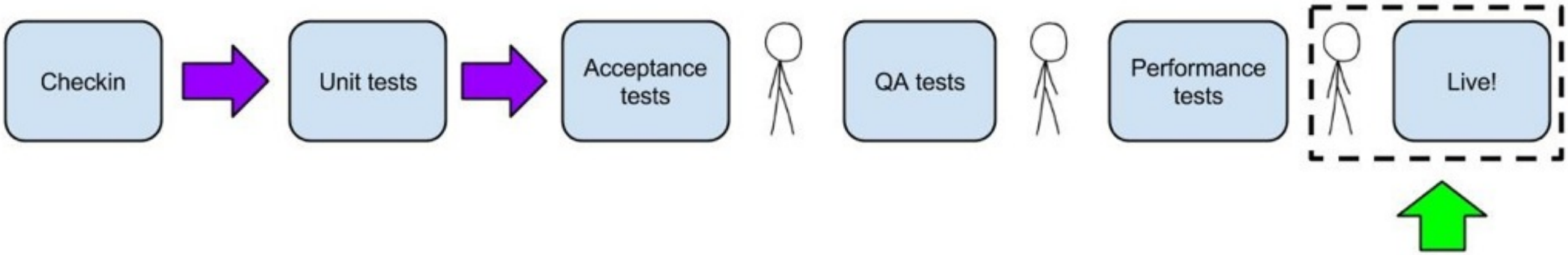
Application Metrics (4 hours)



Rollbacks

- The risk is much higher when rollback isn't possible
- Rollbacks should be normal operation, not a failure
- Users don't care about your new feature if the site is down

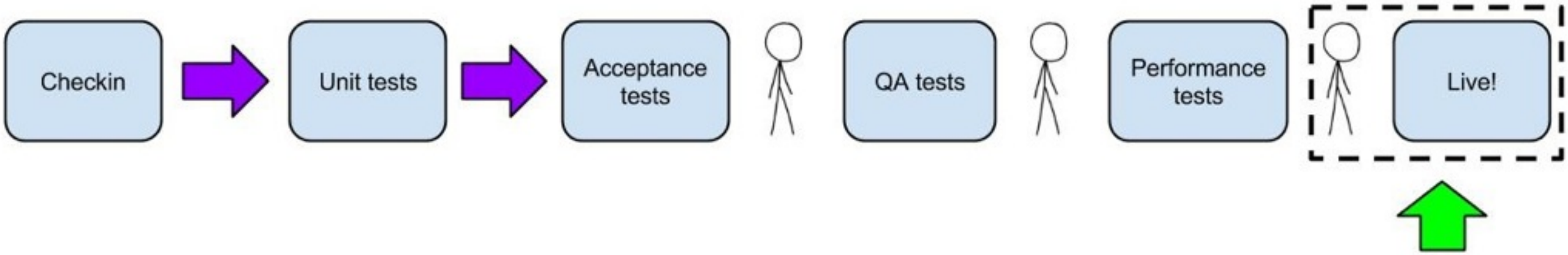
1. It should be possible to rollback
2. It should be quick to rollback



Rollbacks

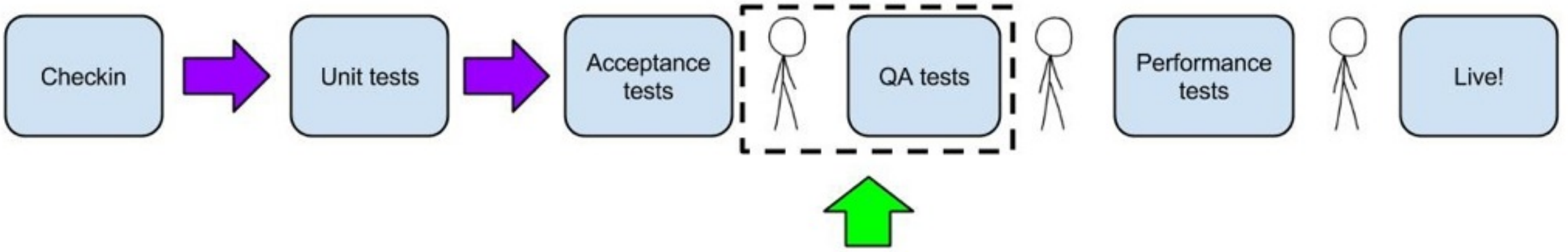
○	4 instances | 8.33 MBytes | java	No	1297 days, 9:51:45 ago by lisa.vangelder@gmail.com	Delete
○	5 instances | 8.63 MBytes | java	No	1297 days, 8:11:34 ago by lisa.vangelder@gmail.com	Delete
○	6 instances | 8.63 MBytes | java	No	1297 days, 3:37:17 ago by lisa.vangelder@gmail.com	Delete
○	7 instances | 8.64 MBytes | java	No	1296 days, 8:00:52 ago by lisa.vangelder@gmail.com	Delete
◉	8 instances | 8.64 MBytes | java	**Yes**	1296 days, 7:57:43 ago by lisa.vangelder@gmail.com	*Cannot delete default version.*

Make Default



A release should be a non-event

- Done in working hours
- Easy to monitor
- Easy to rollback

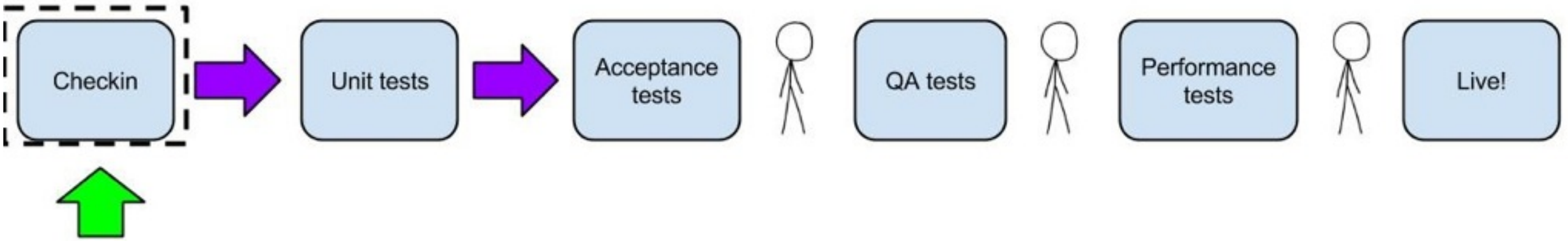


QA don’t have time to test

Dev Capacity: 17

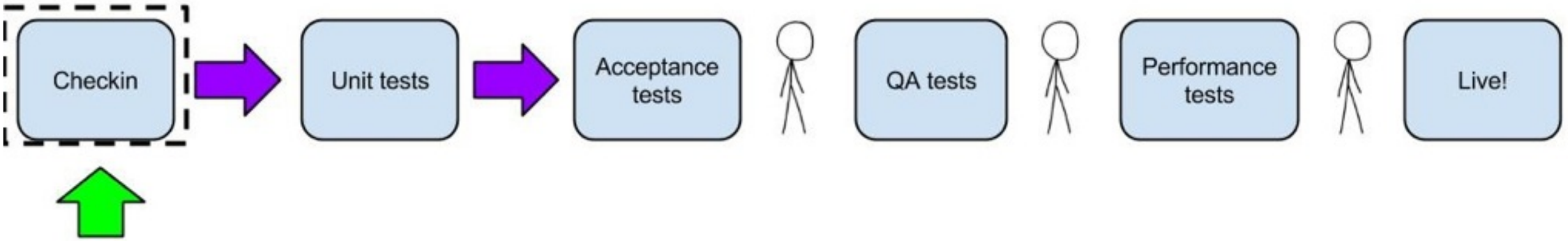
QA Capacity: 6

Dev: 7 QA: 3	Add comment to widget
Dev: 5 QA: 3	Widget sharable on facebook
Dev: 5 QA: 3	Widget sharable on twitter

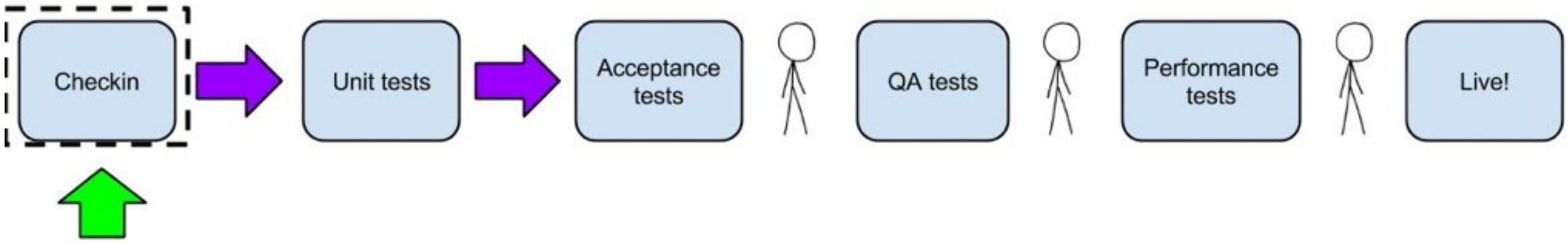


Cross-functional issues

- release waiting on qa sign off
- release waiting on product manager sign off
- developers waiting on designs
- ux waiting on developers
- front-end developers waiting on back-end

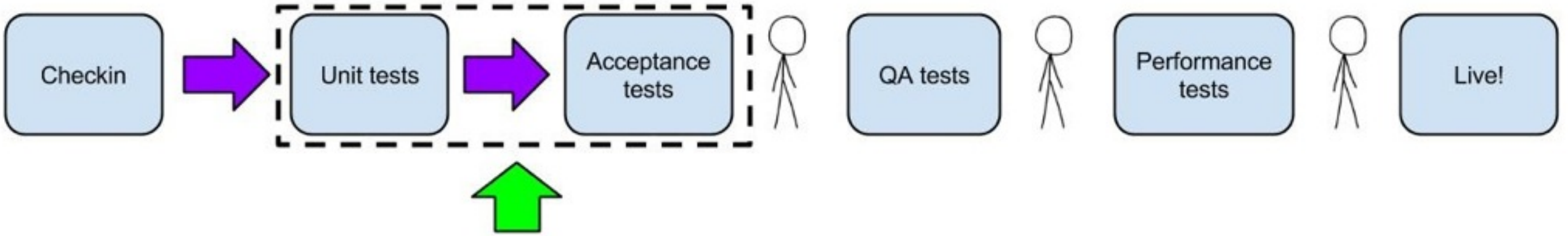


Backlog	In Dev	In QA	Blocked	Ready	Released
Widget feature 9	Widget feature 7	Widget feature 6	Widget feature 3		Widget feature 1
Widget feature 10	Widget feature 8		Widget feature 4		Widget feature 2
Widget feature 11			Widget feature 5		



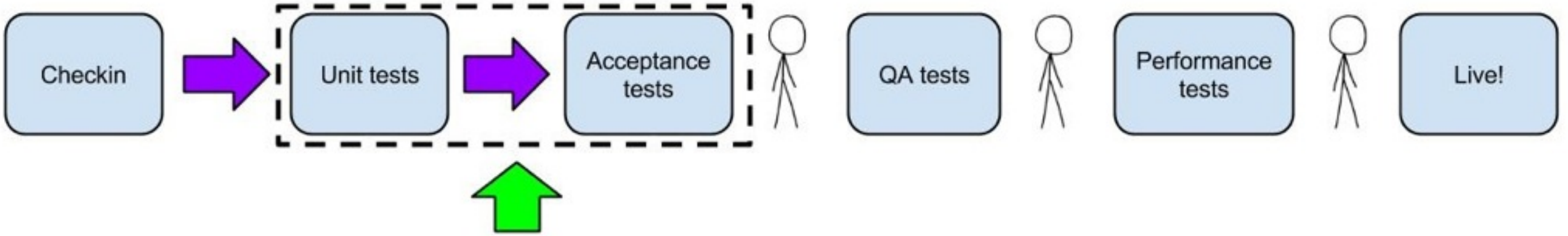
Cross functional teams

- don’t start story if all resources aren’t available
- blockers should block
- when work is held up - can someone else perform that function?



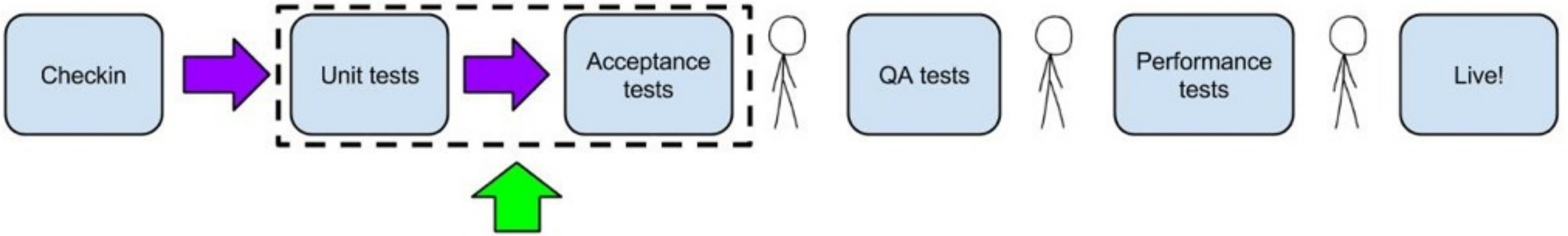
Takes too long to get a green build

- Flaky tests
- Slow-running tests
- Merge hell



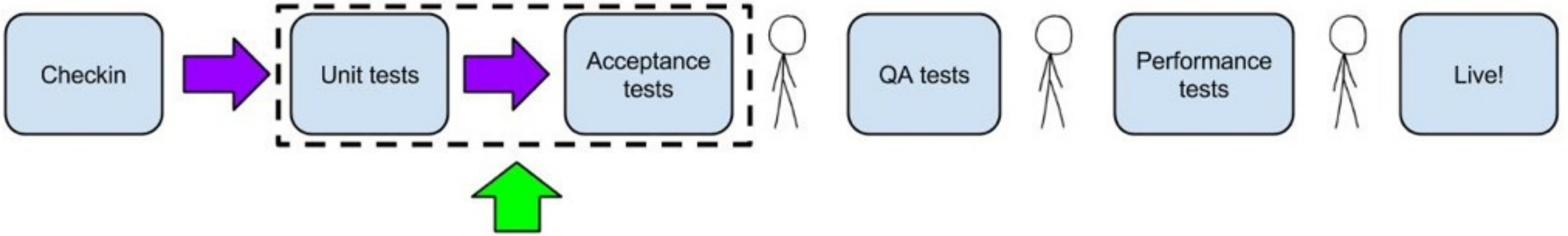
Flaky tests

- The tests that cry wolf
- Isolate them
- Fix them or delete them



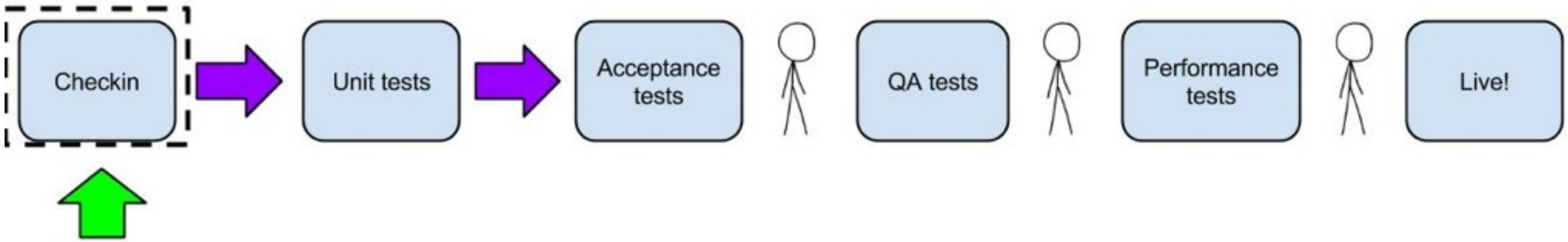
Slow running tests

- More than 5 minutes is slow
- Waste of developer time
- Interrupt flow to fix
- People deploy without waiting for tests
- Frequent broken build



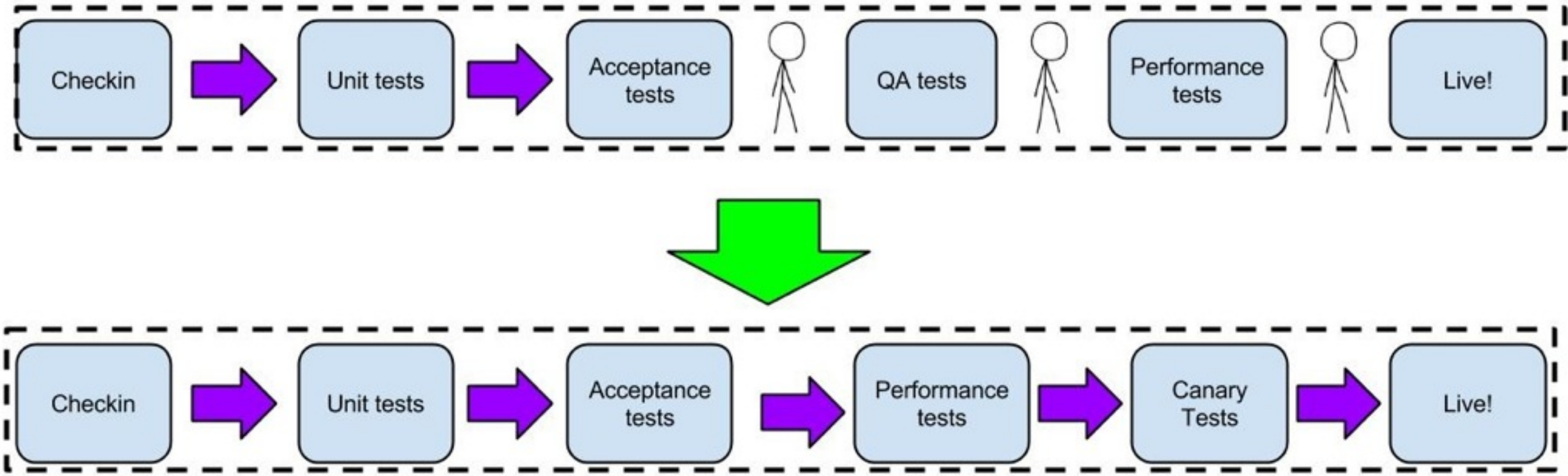
Slow running tests

- separate unit tests from acceptance tests
- limit the amount of acceptance tests
- mock dependencies - limit calls to db
- run tests in parallel

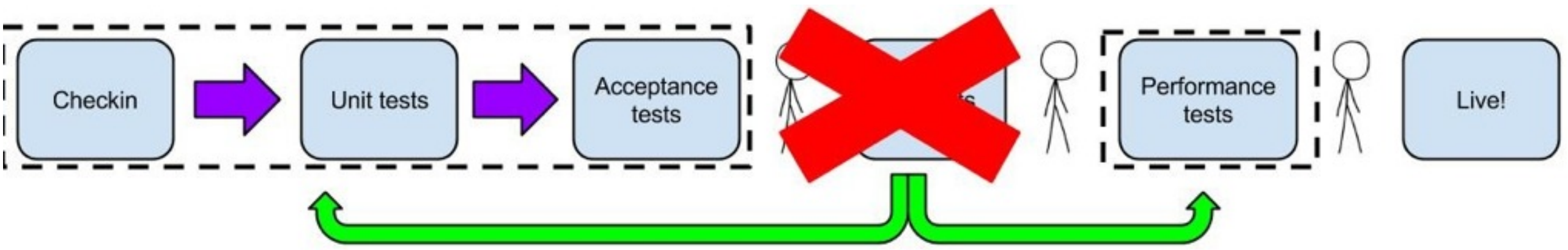


Merge hell

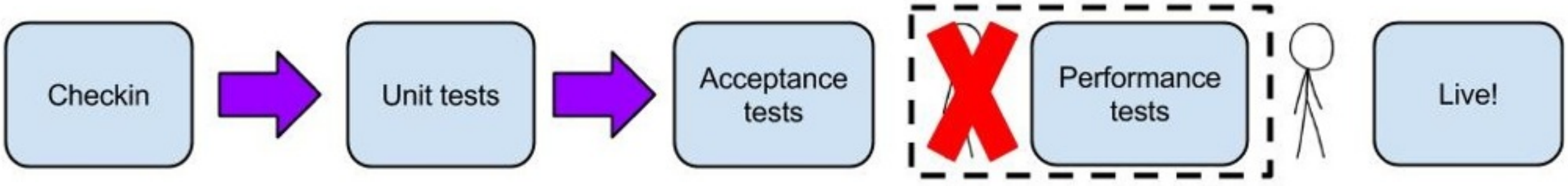
- Continuous Integration is more often than you think!
- Don’t have long lived feature branches
- Check in to master at least once a day
- Feature switches
- Branch by abstraction


Checkin
Unit tests
Acceptance tests
QA tests
Performance tests
Live!
Checkin
Unit tests
Acceptance tests
Performance tests
Canary Tests
Live!

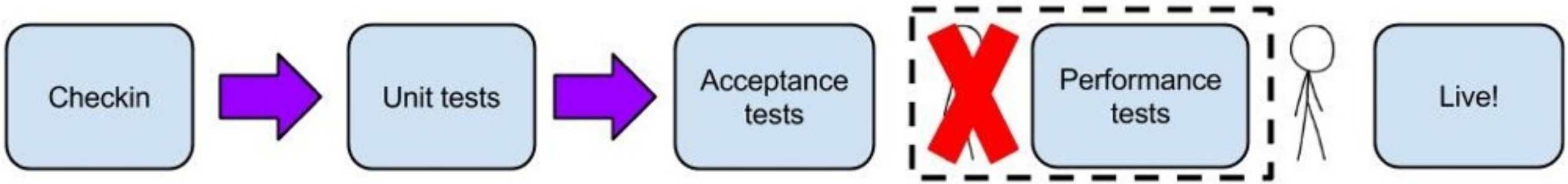
QA define automated testing strategy



Automate performance tests



What criteria do humans use to evaluate performance tests?



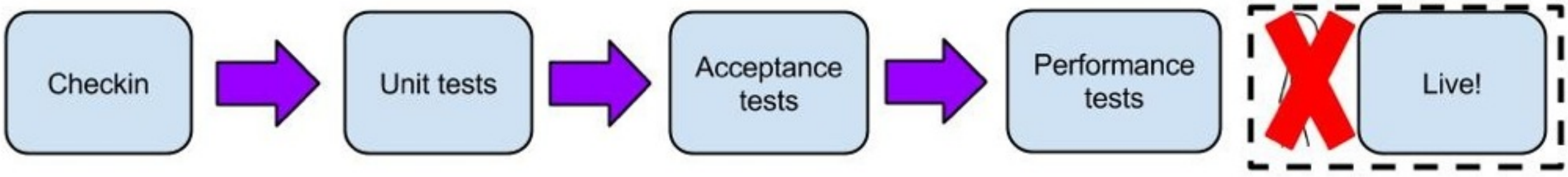
- Have a performance test environment that is a scaled-down replica of production

- Automate log collection, make sure tests reflect current traffic patterns.

- Use your application-specific metrics

- Define acceptable ranges for your application

Automate release process



What criteria do humans use to evaluate a successful release?

Use your application-specific metrics and acceptable ranges

Summary

- Measure your cycle time
- Fix your bottlenecks
- Improve your reaction time

Questions?

Feedback please!

http://bit.ly/1q3FFIo

Suggestions for further reading

- Continuous Delivery by Jez Humble & David Farley
- The Phoenix Project by Gene Kim, Kevin Behr & George Spafford
- The Goal by Eliyahu M. Goldratt & Jeff Cox
- Lean Software Development by Mary Poppendieck & Tom Poppendieck
- Release It! by Michael T. Nygard