DPE
SUMMIT

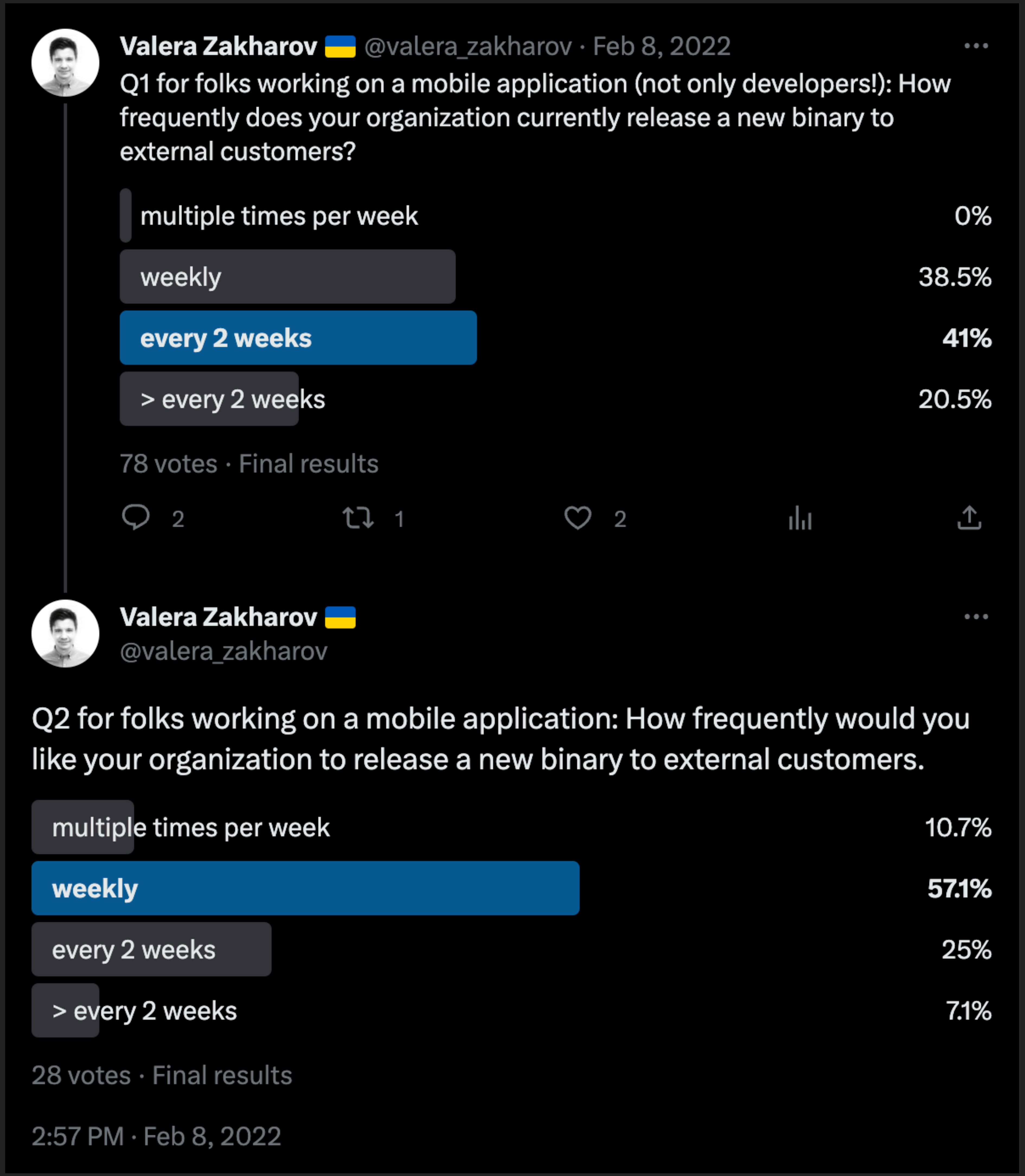
#’S ROAD TO FASTER MOBILE PRODUCT ITERATIONS

FASTER. ~~HARDER. STRONGER~~ FREQUENTER.

Spend $$$ on AI to save devs a few minutes per day
Simple process change to ship features 2x faster
DPE IN 2023
JAKE-CLARK.TUMBLR

ONE DOES NOT SIMPLY
MAKE A RELEASE PROCESS CHANGE

WE ARE HERE TO HELP WITH SHIPPING FEATURES TO USERS



Valera Zakharov 🇺🇦 @valera_zakharov · Feb 8, 2022

Q1 for folks working on a mobile application (not only developers!): How frequently does your organization currently release a new binary to external customers?

multiple times per week	0%
weekly	38.5%
every 2 weeks	**41%**
> every 2 weeks	20.5%

78 votes · Final results

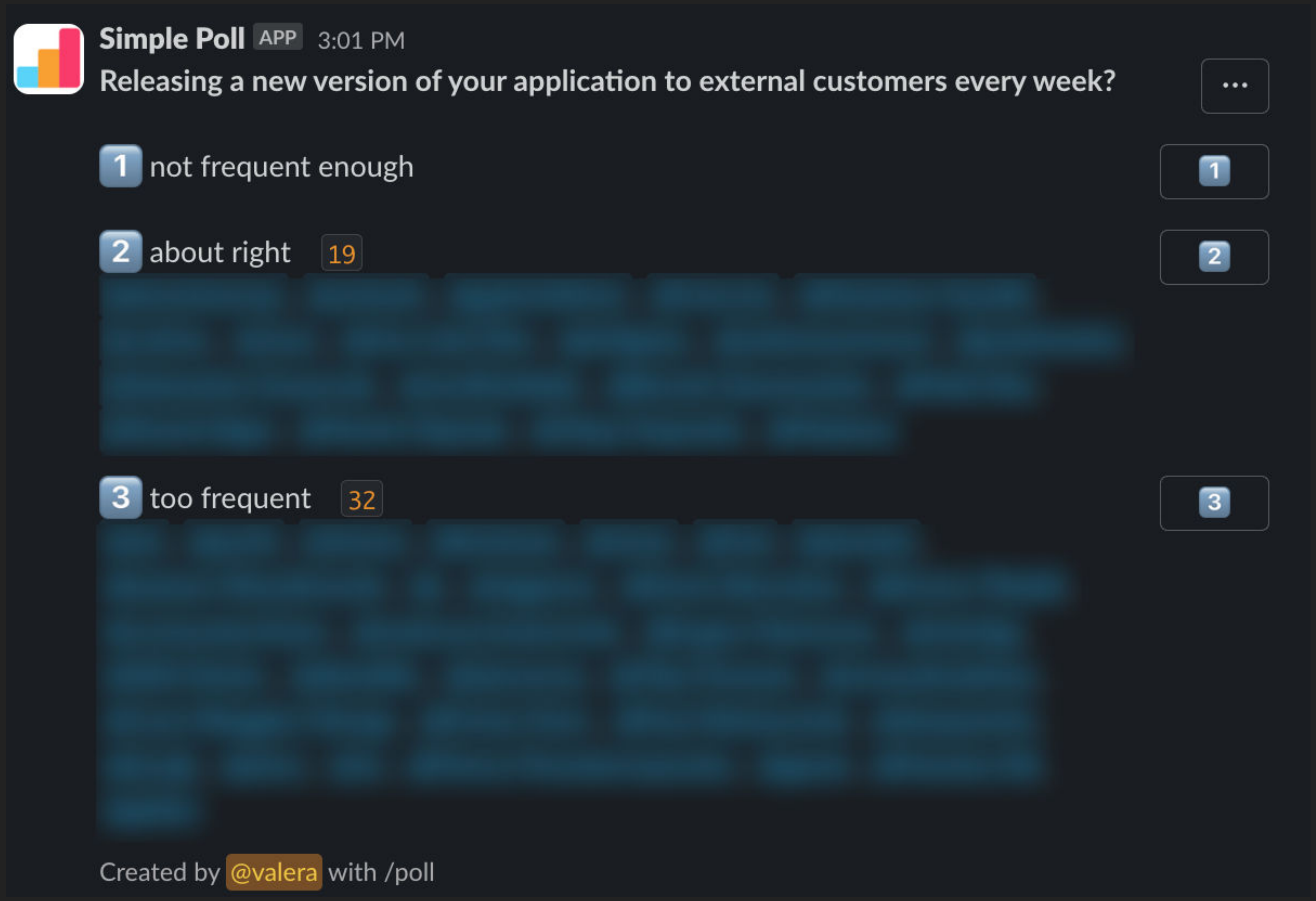
Valera Zakharov 🇺🇦
@valera_zakharov

Q2 for folks working on a mobile application: How frequently would you like your organization to release a new binary to external customers.

multiple times per week	10.7%
weekly	**57.1%**
every 2 weeks	25%
> every 2 weeks	7.1%

28 votes · Final results

2:57 PM · Feb 8, 2022



Simple Poll APP 3:01 PM

Releasing a new version of your application to external customers every week?

1 not frequent enough

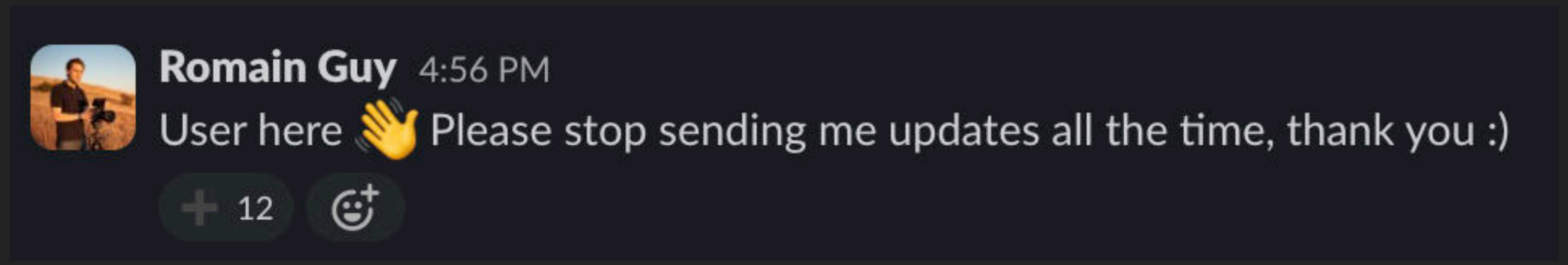
2 about right 19

[redacted]

3 too frequent 32

[redacted]

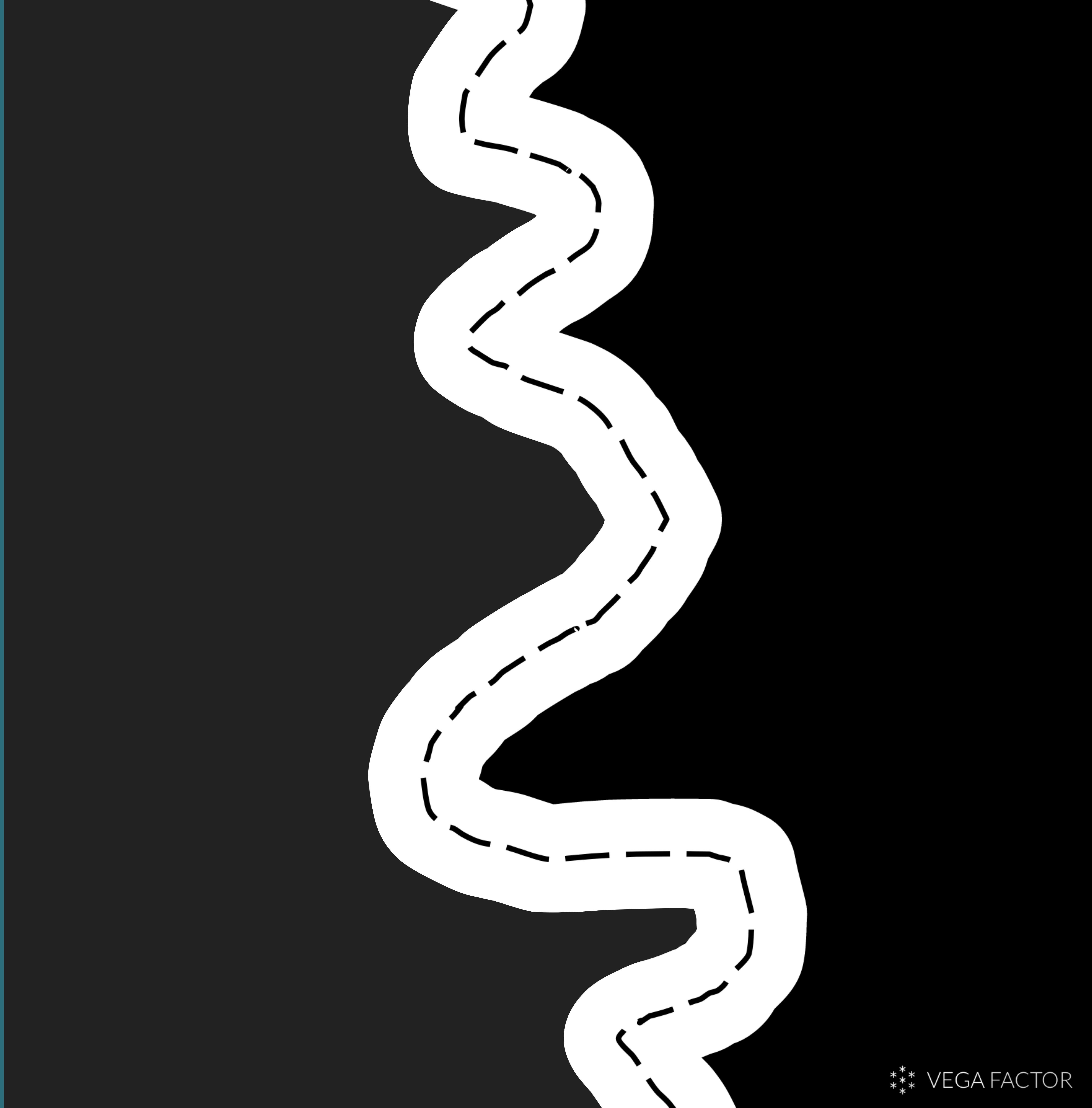
Created by @valera with /poll



Romain Guy 4:56 PM

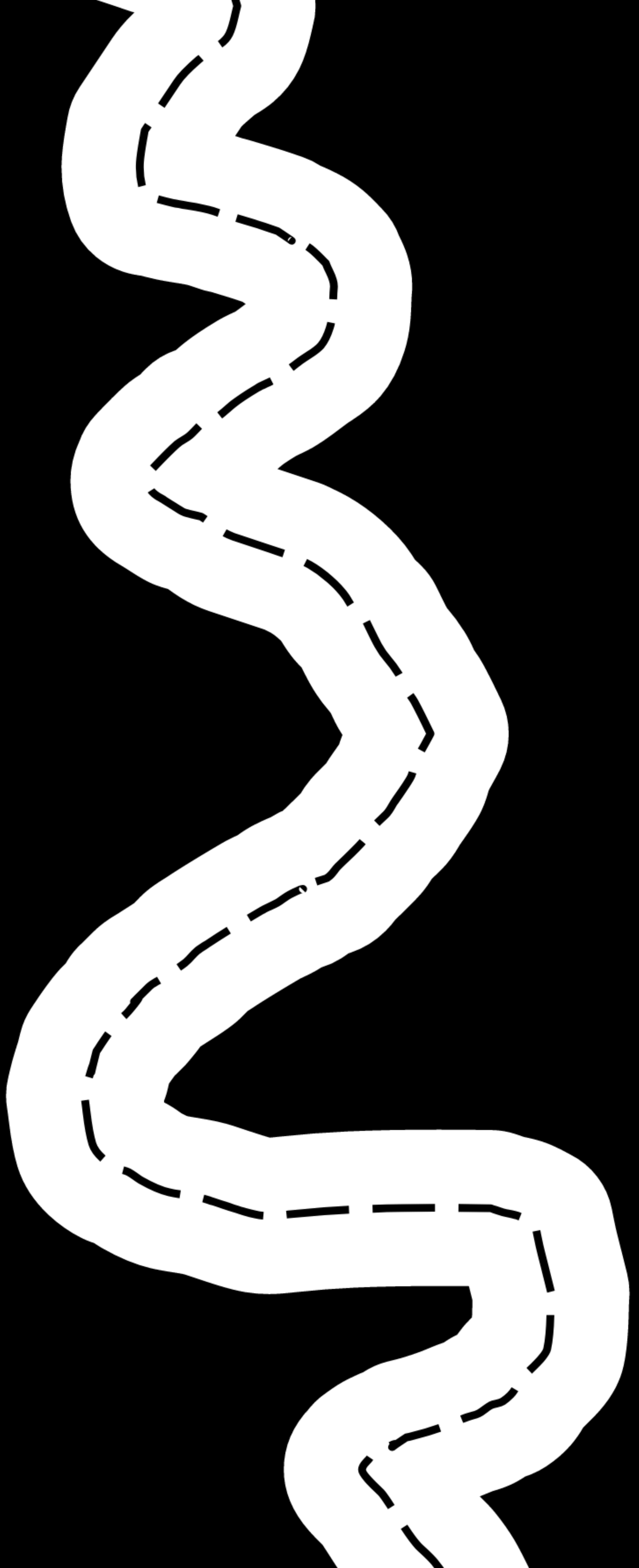
User here 👋 Please stop sending me updates all the time, thank you :)

Imagine you were driving a car down a windy road. If you could only touch the steering wheel once every ten minutes, how fast would you drive?



VEGA FACTOR

A team's speed is a function of the frequency and quality of its feedback loops



https://www.independent.co.uk/travel/news-and-advice/japan-s-bullet-train-50-years-of-powering-ahead-9940290.html

MIN ITERATION CYCLE ON WEB

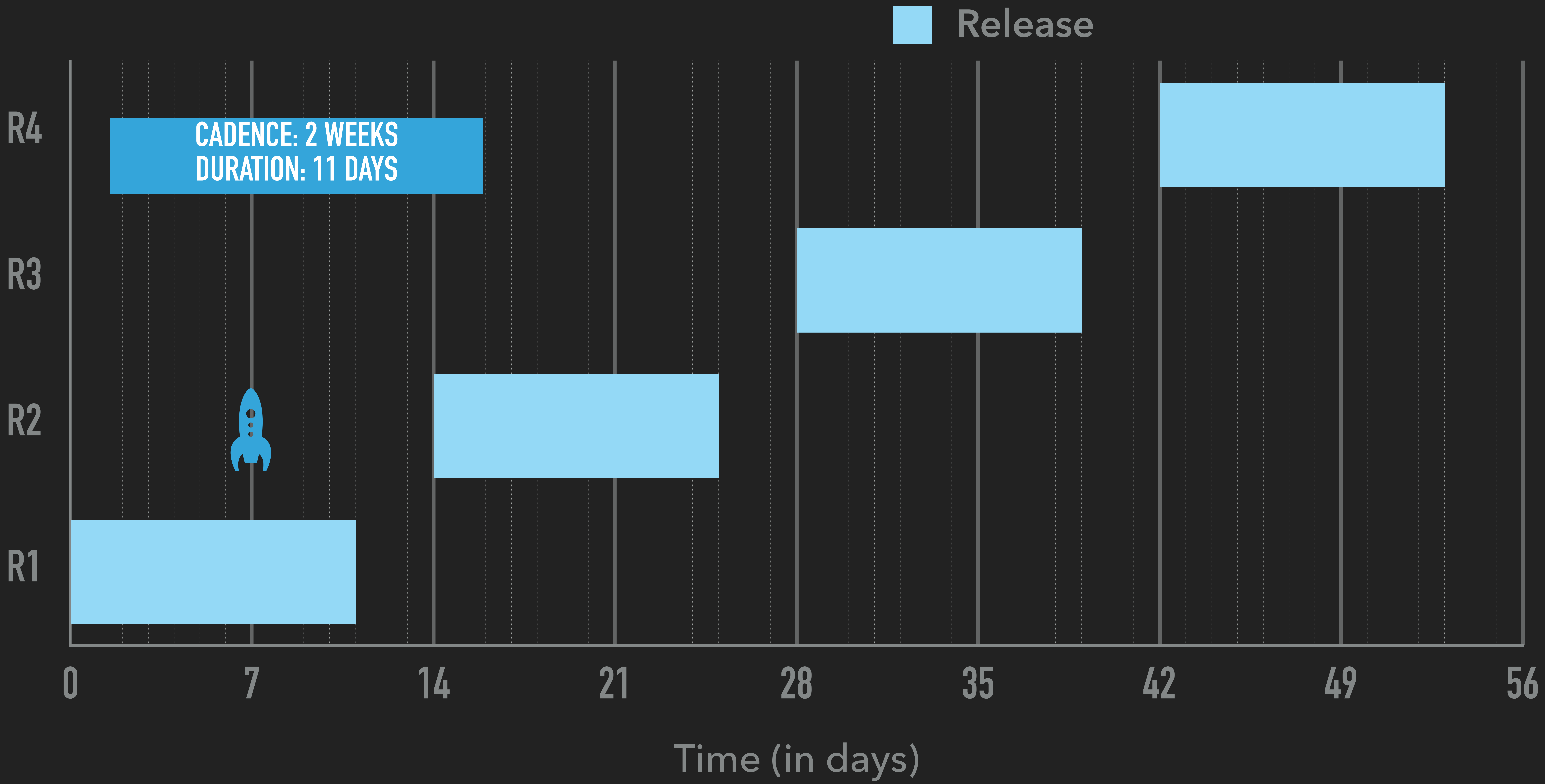
~1 DAY

https://www.bbc.com/travel/article/20230313-the-slowest-train-journey-in-india

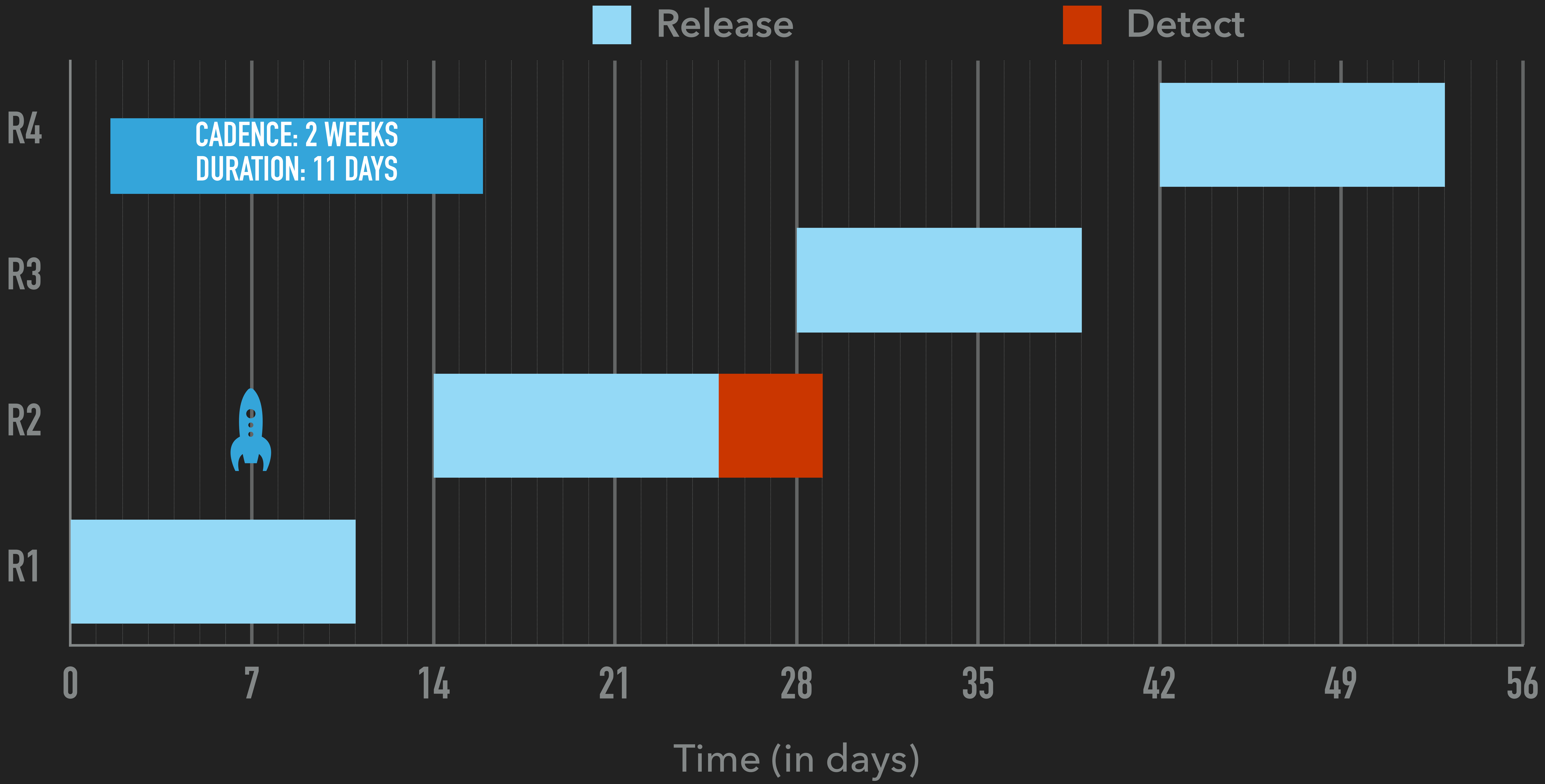
MIN ITERATION ON MOBILE*

40+ DAYS

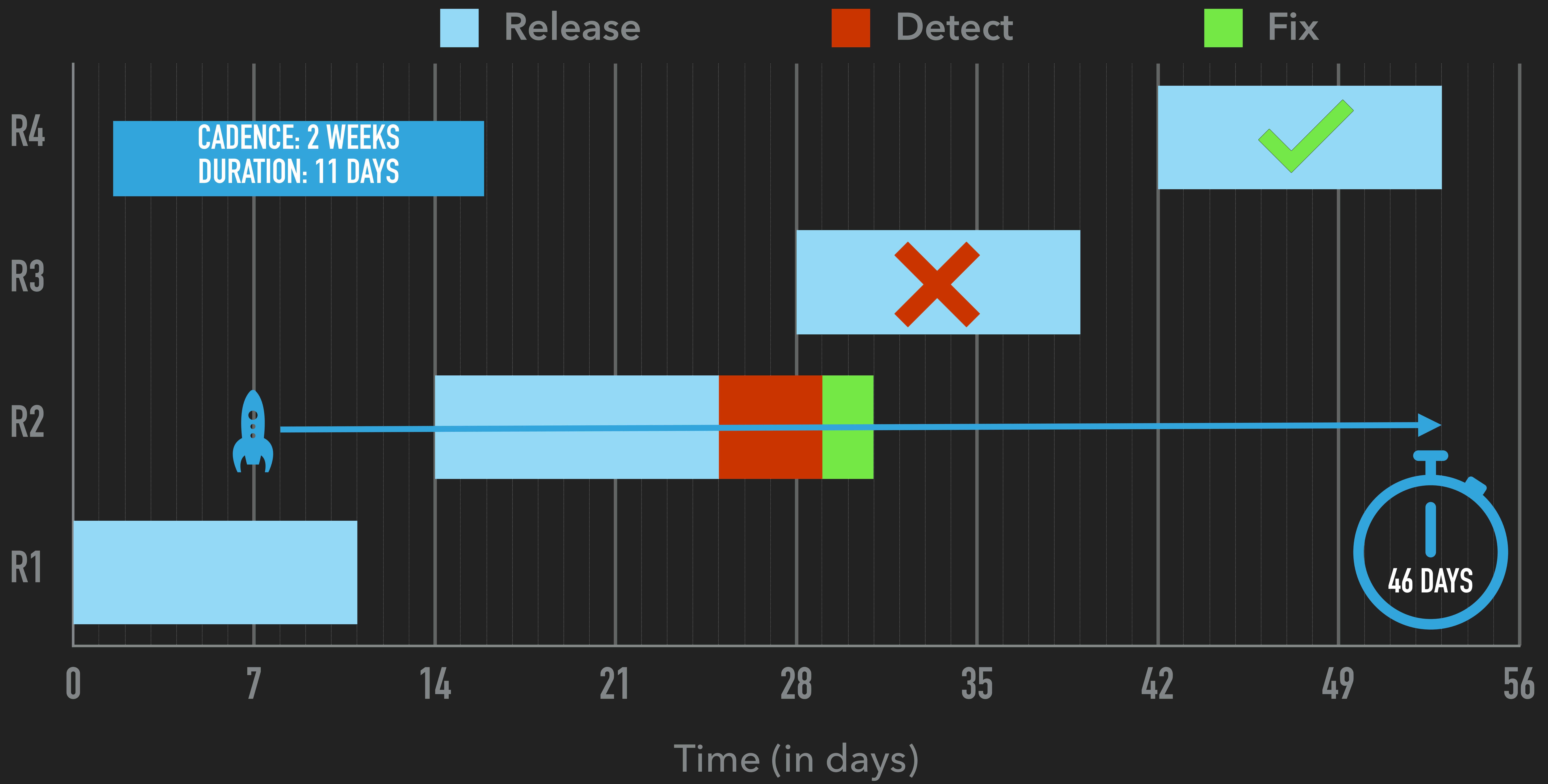
*Slack Apps in 2022



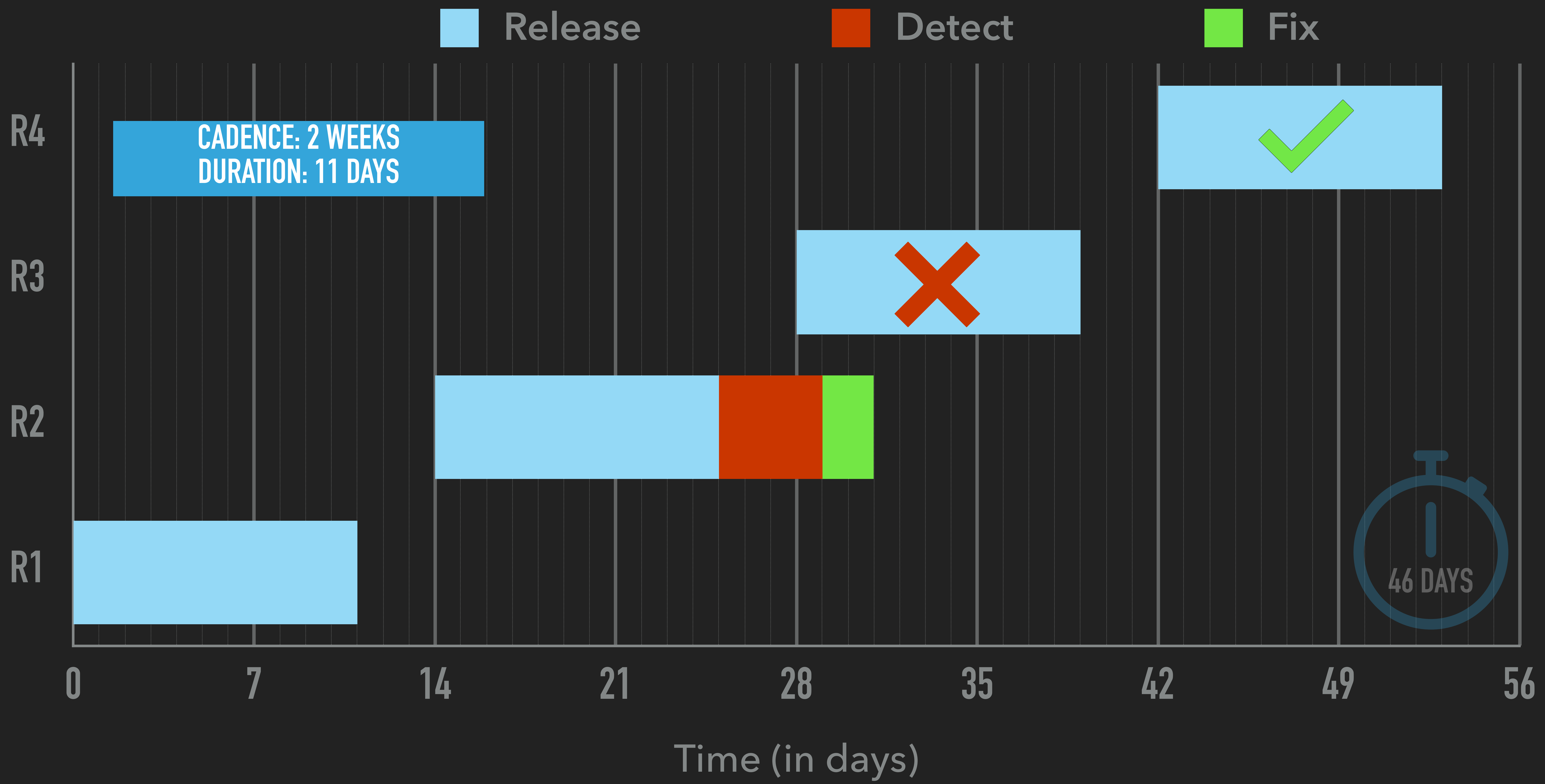
CADENCE: 2 WEEKS
DURATION: 11 DAYS

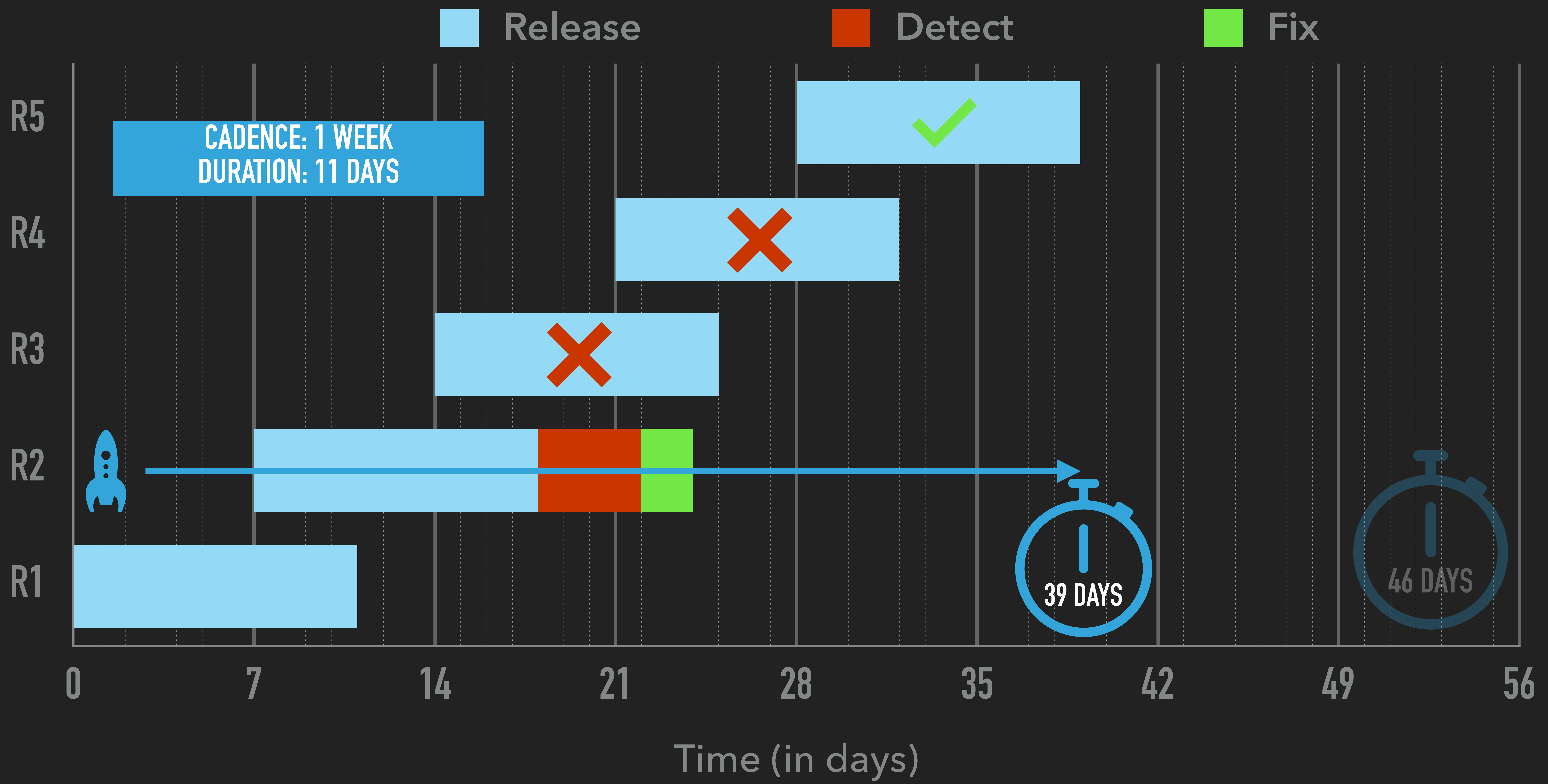


CADENCE: 2 WEEKS
DURATION: 11 DAYS

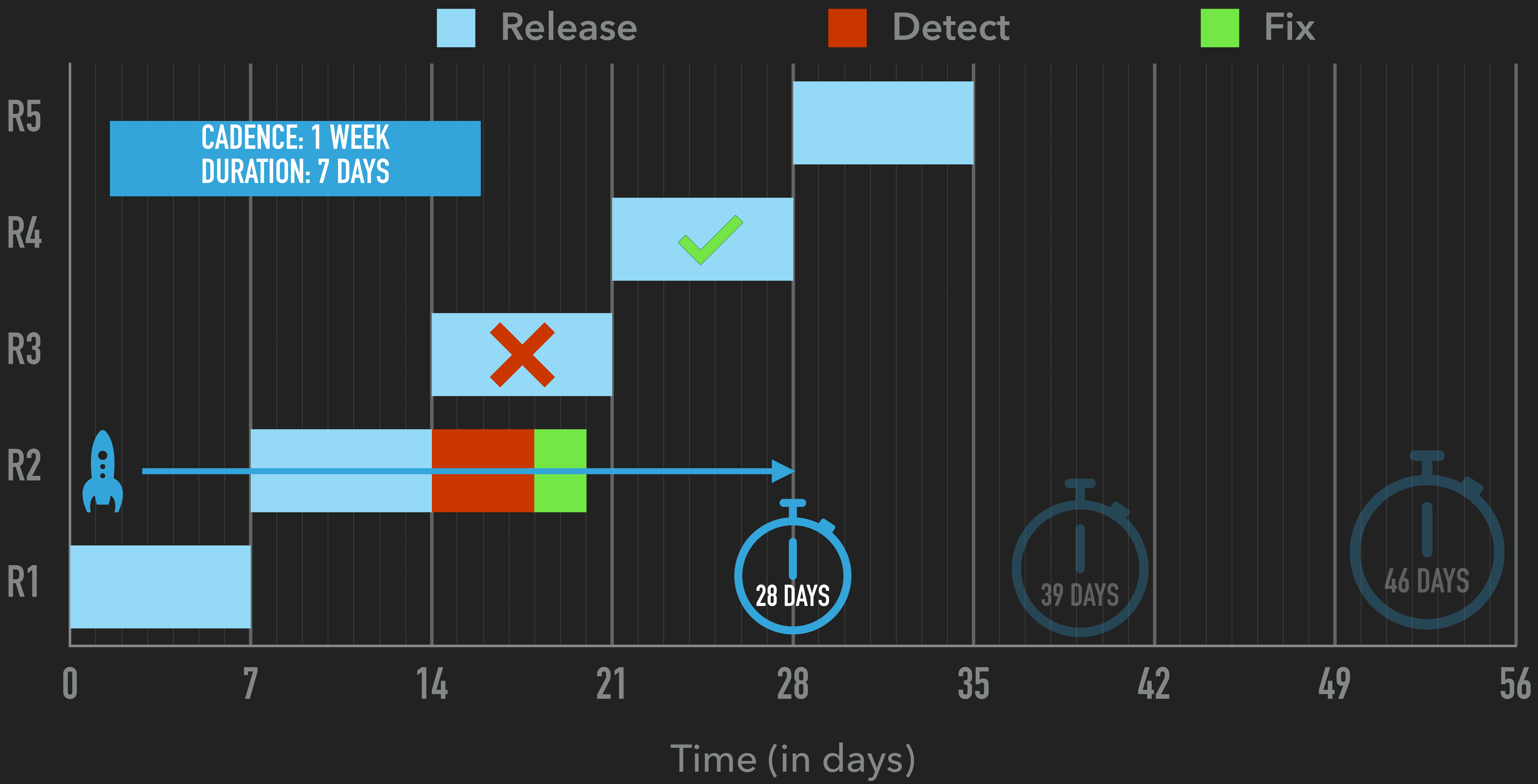


Release
Detect
Fix
R4
R3
R2
R1
CADENCE: 2 WEEKS
DURATION: 11 DAYS
46 DAYS
0
7
14
21
28
35
42
49
56
Time (in days)


Release
Detect
Fix
R4
R3
R2
R1
CADENCE: 2 WEEKS
DURATION: 11 DAYS
46 DAYS
0
7
14
21
28
35
42
49
56
Time (in days)

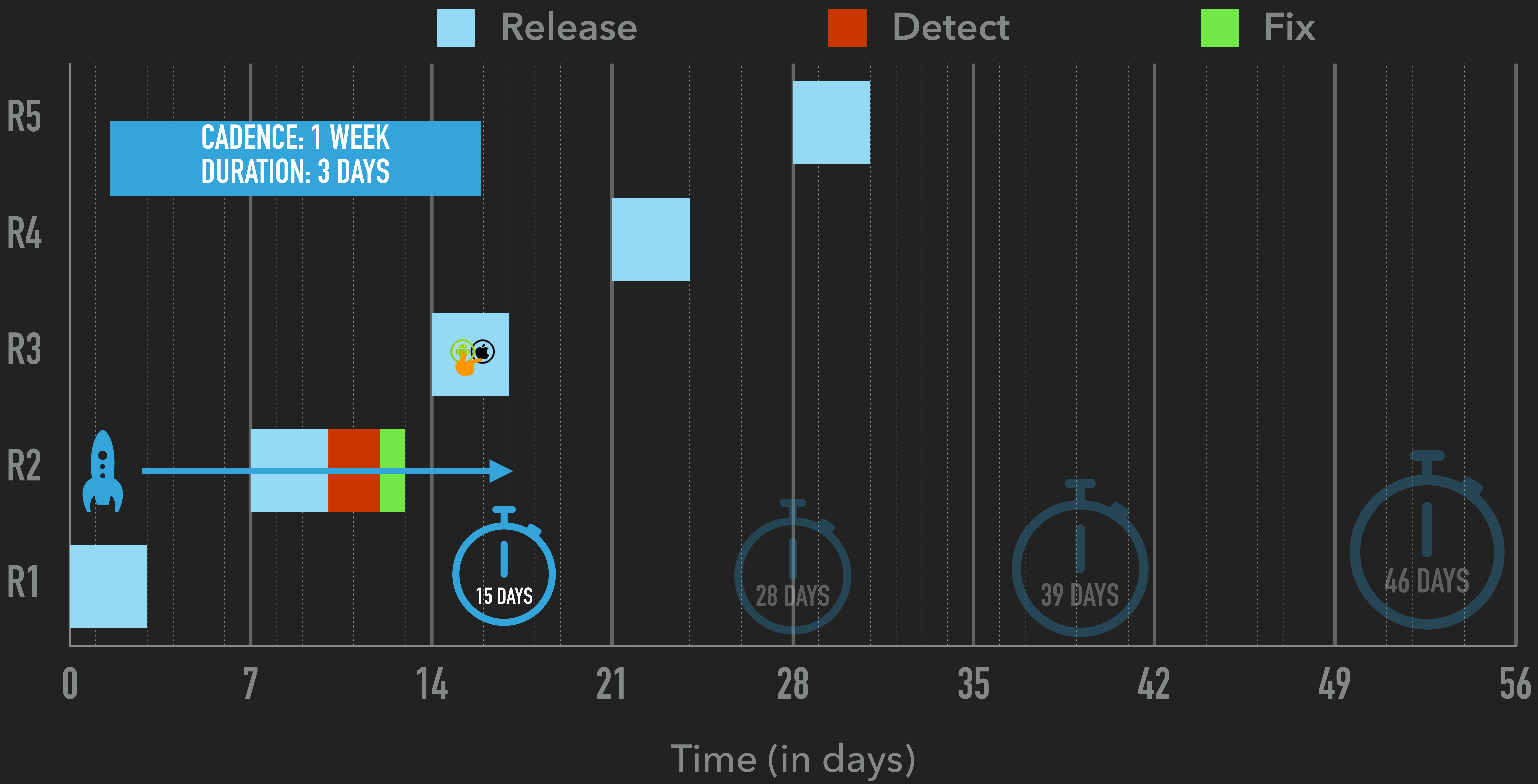


Release
Detect
Fix
R5
R4
R3
R2
R1
CADENCE: 1 WEEK
DURATION: 11 DAYS
39 DAYS
46 DAYS
0
7
14
21
28
35
42
49
56
Time (in days)

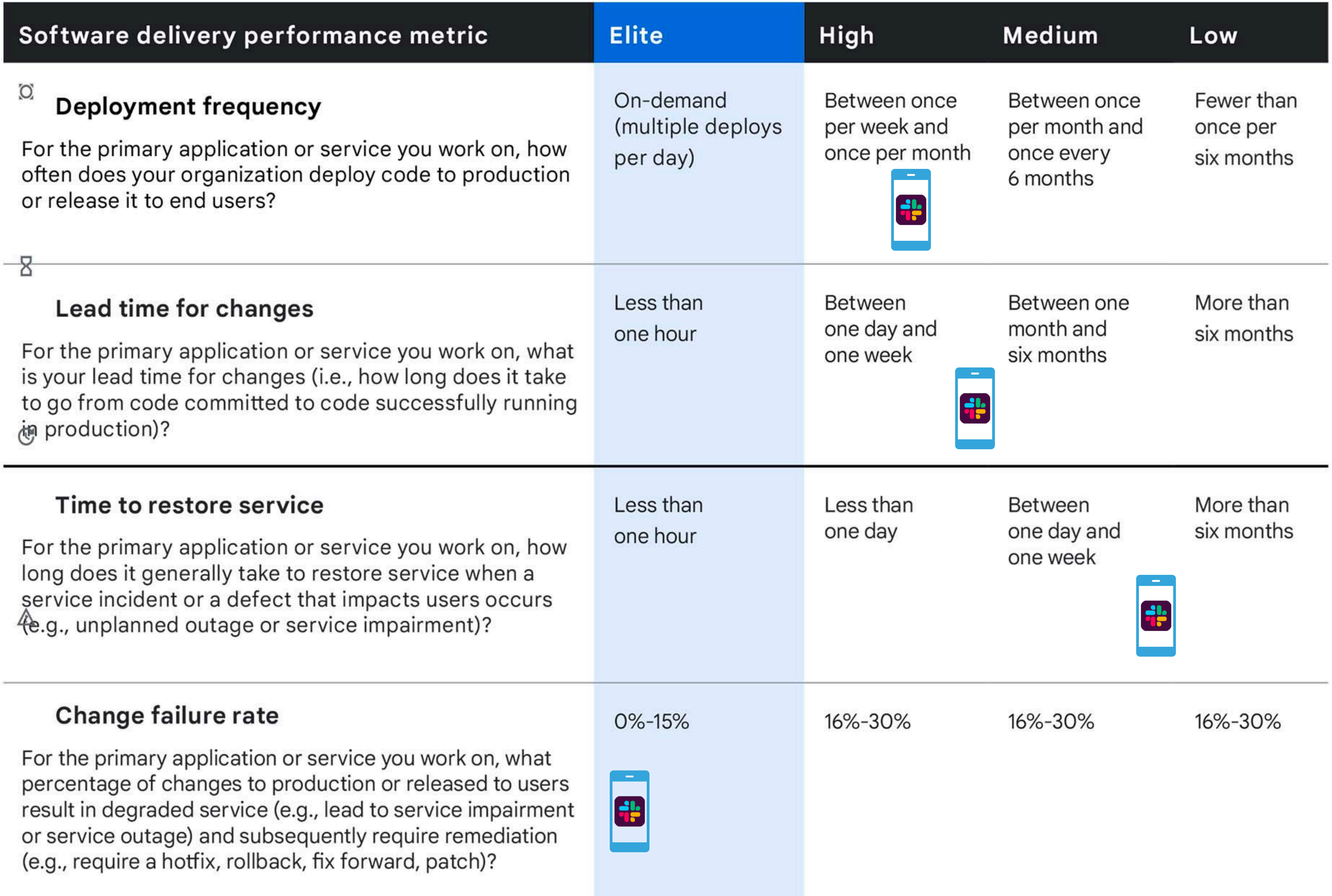


Release
Detect
Fix
R5
R4
R3
R2
R1
CADENCE: 1 WEEK
DURATION: 7 DAYS
28 DAYS
39 DAYS
46 DAYS
0
7
14
21
28
35
42
49
56
Time (in days)

RELEASE MORE FREQUENTLY OR FASTER?

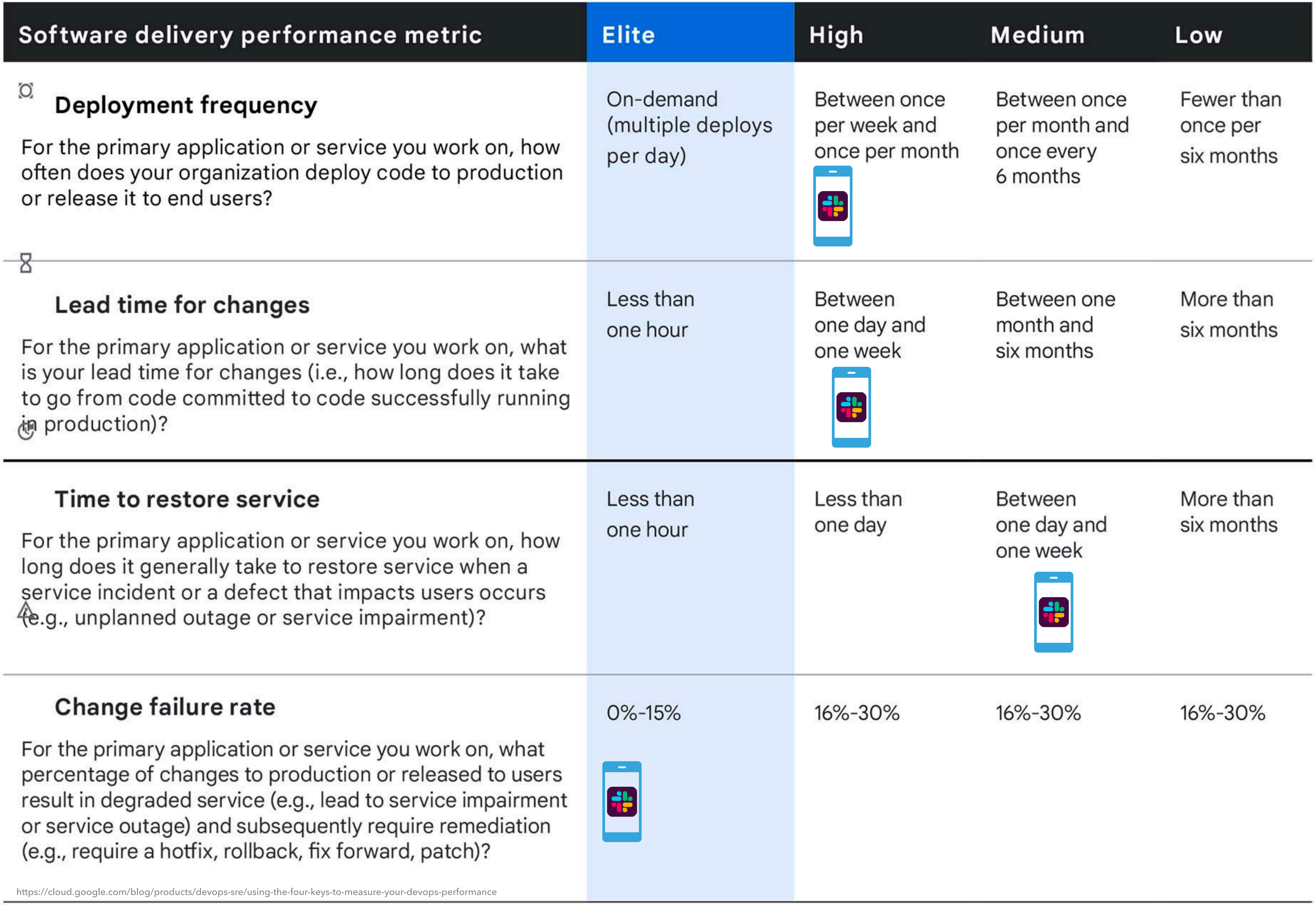


CADENCE: 1 WEEK
DURATION: 3 DAYS



Software delivery performance metric	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per week and once per month	Between once per month and once every 6 months	Fewer than once per six months
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one hour	Between one day and one week	Between one month and six months	More than six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day	Between one day and one week	More than six months
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0%-15%	16%-30%	16%-30%	16%-30%

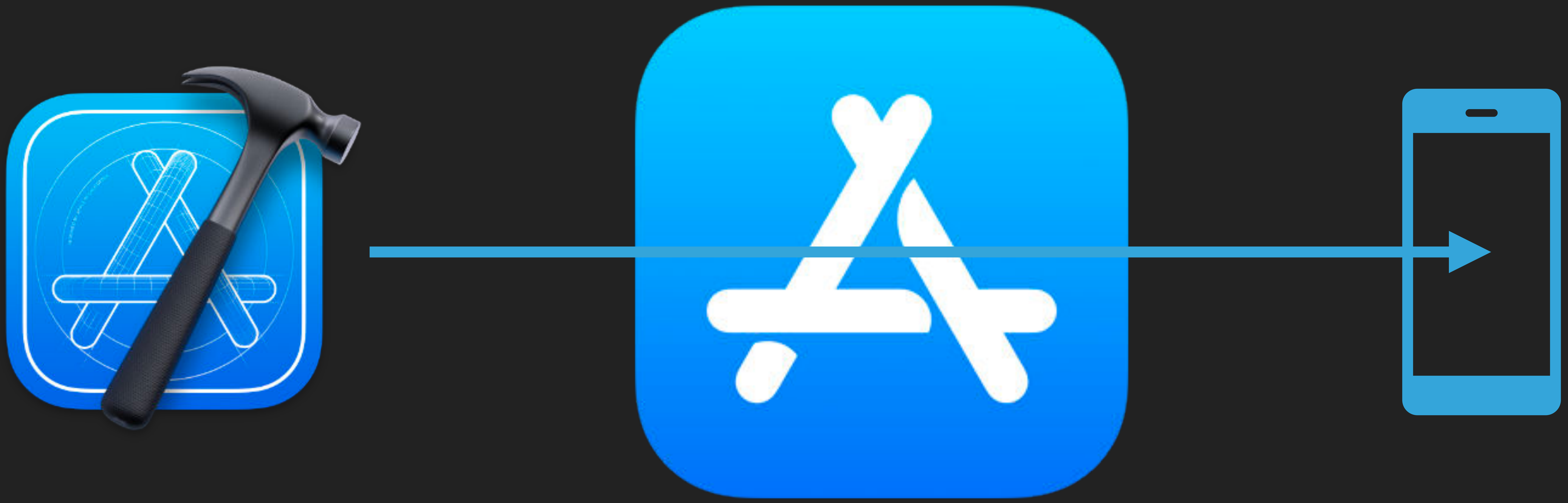
https://cloud.google.com/blog/products/devops-sre/using-the-four-keys-to-measure-your-devops-performance



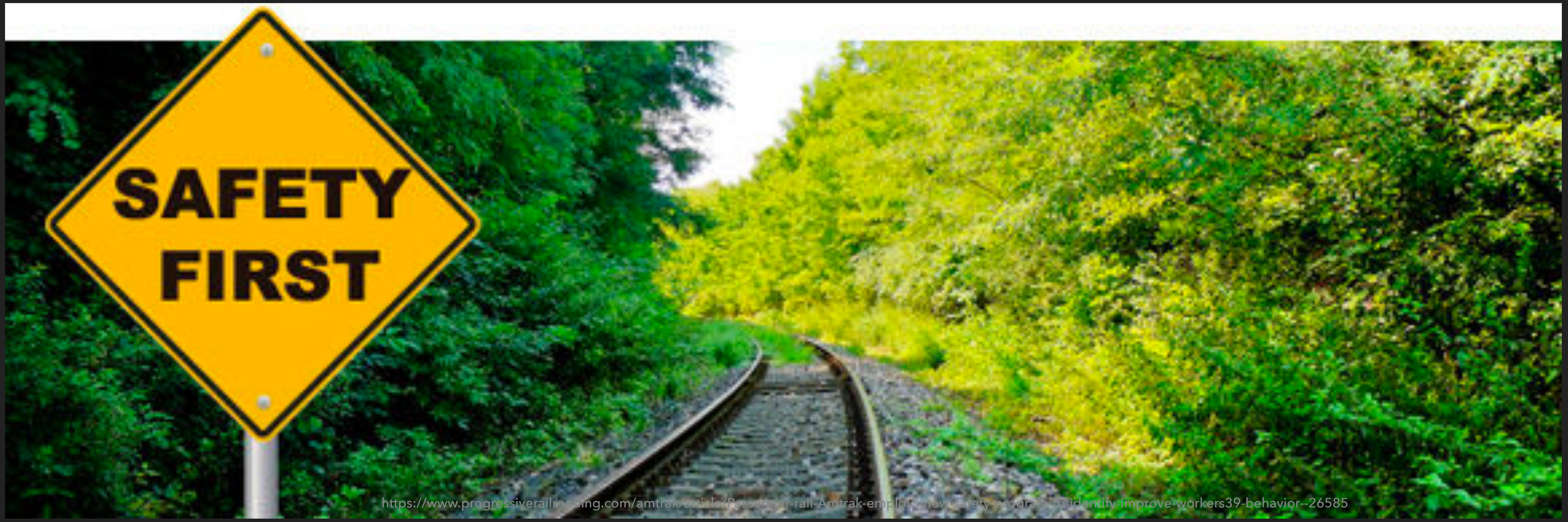
Software delivery performance metric	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per week and once per month 📱	Between once per month and once every 6 months	Fewer than once per six months
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one hour	Between one day and one week 📱	Between one month and six months	More than six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day	Between one day and one week 📱	More than six months
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0%-15% 📱	16%-30%	16%-30%	16%-30%

https://cloud.google.com/blog/products/devops-sre/using-the-four-keys-to-measure-your-devops-performance

MOBILE IS DIFFERENT



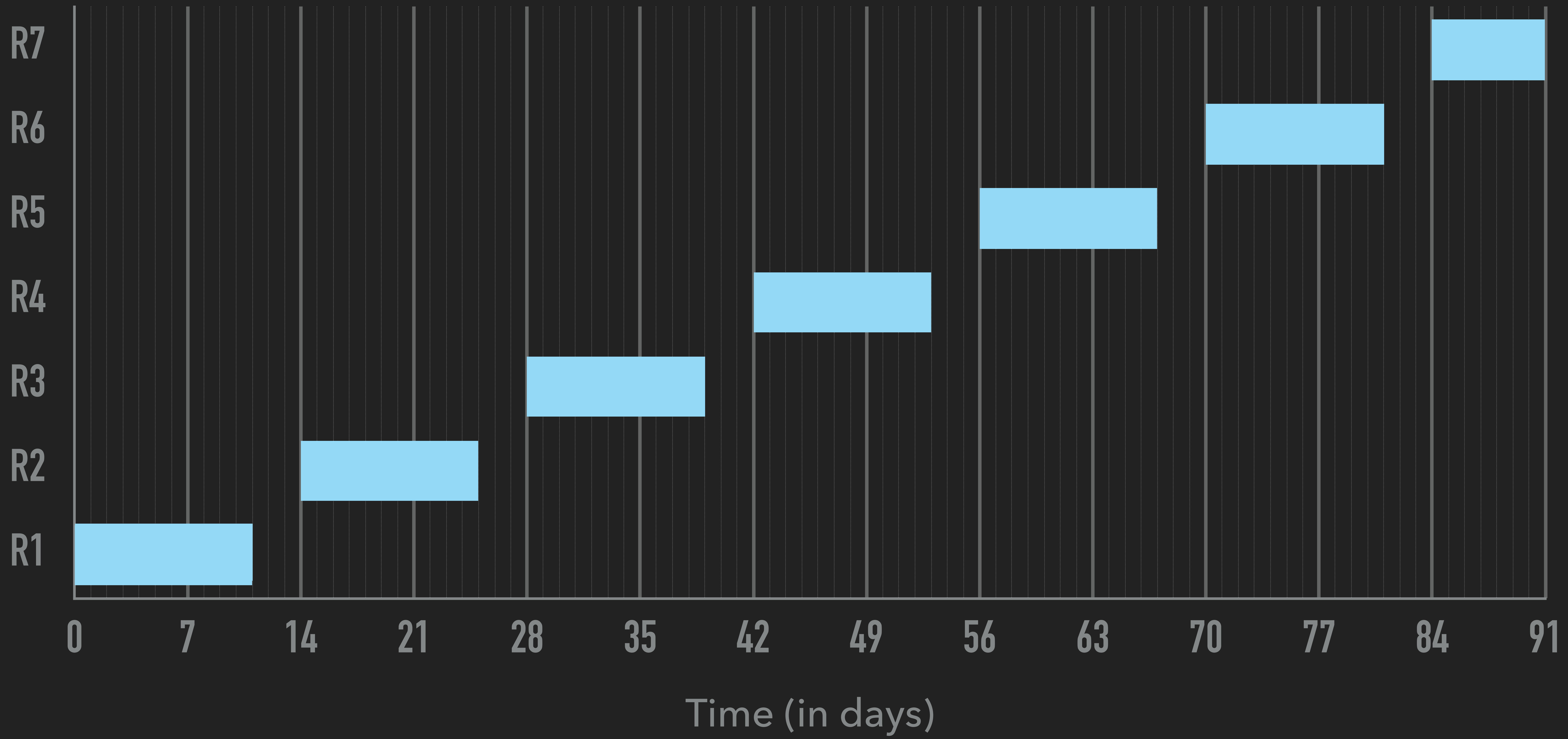
MOBILE IS DIFFERENT



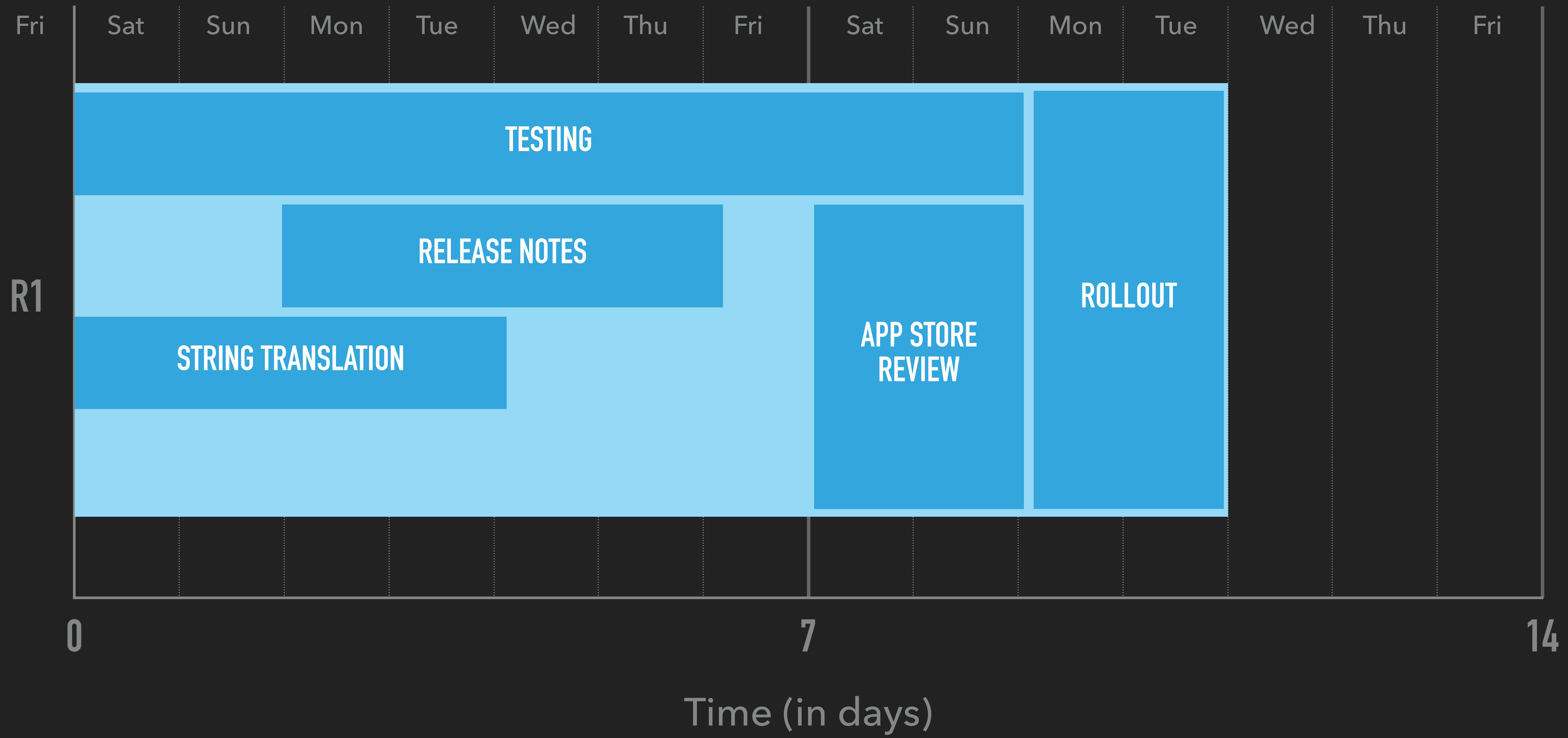
https://www.progressiverailroading.com/amtrak/article/Passenger-rail-Amtrak-employs-new-safety-program-to-identify-improve-workers39-behavior--26585

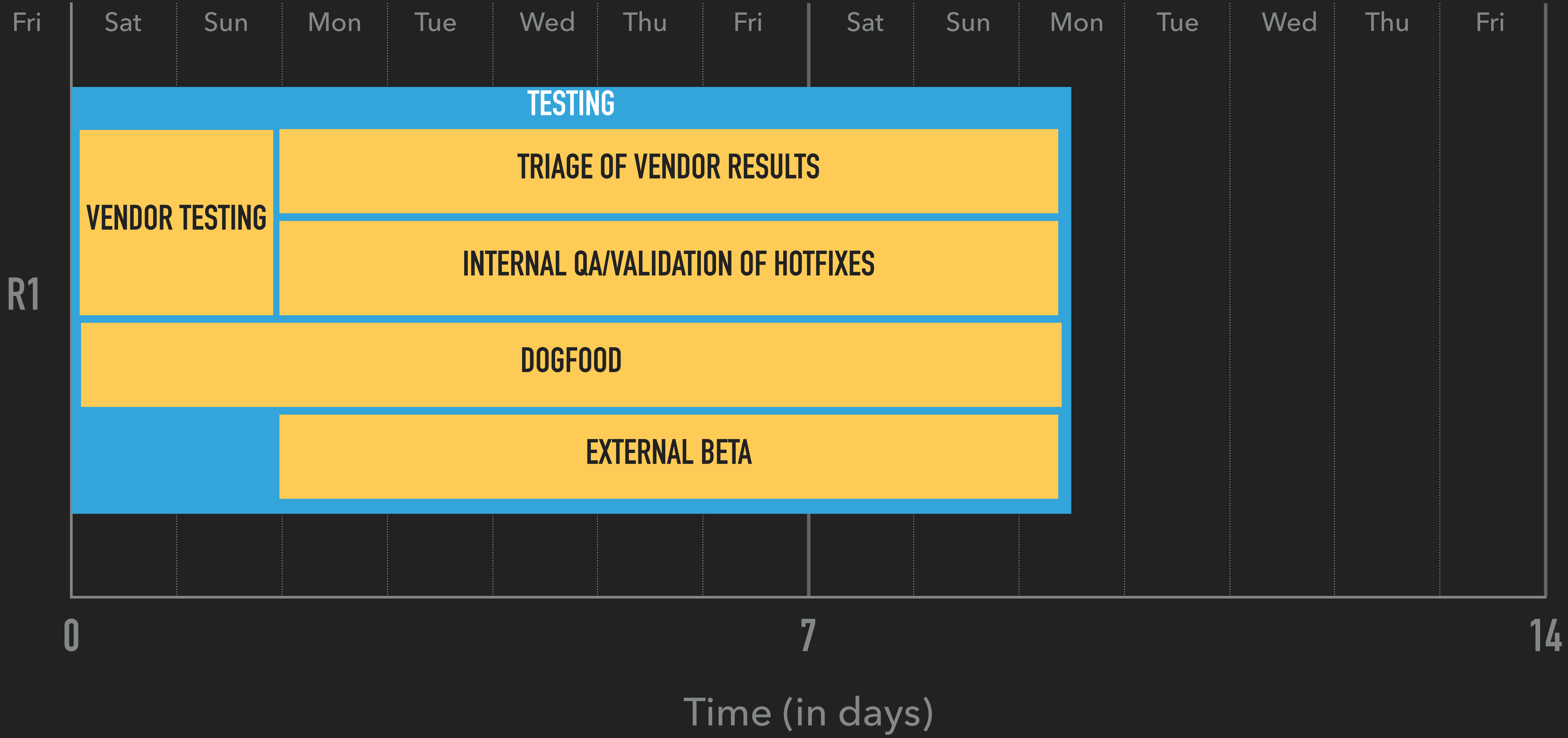
- App Store review takes days. Sometimes multiple rounds.
- No fast rollbacks
- Users not guaranteed to install your update right away or ever
- Cost of a mistake is high

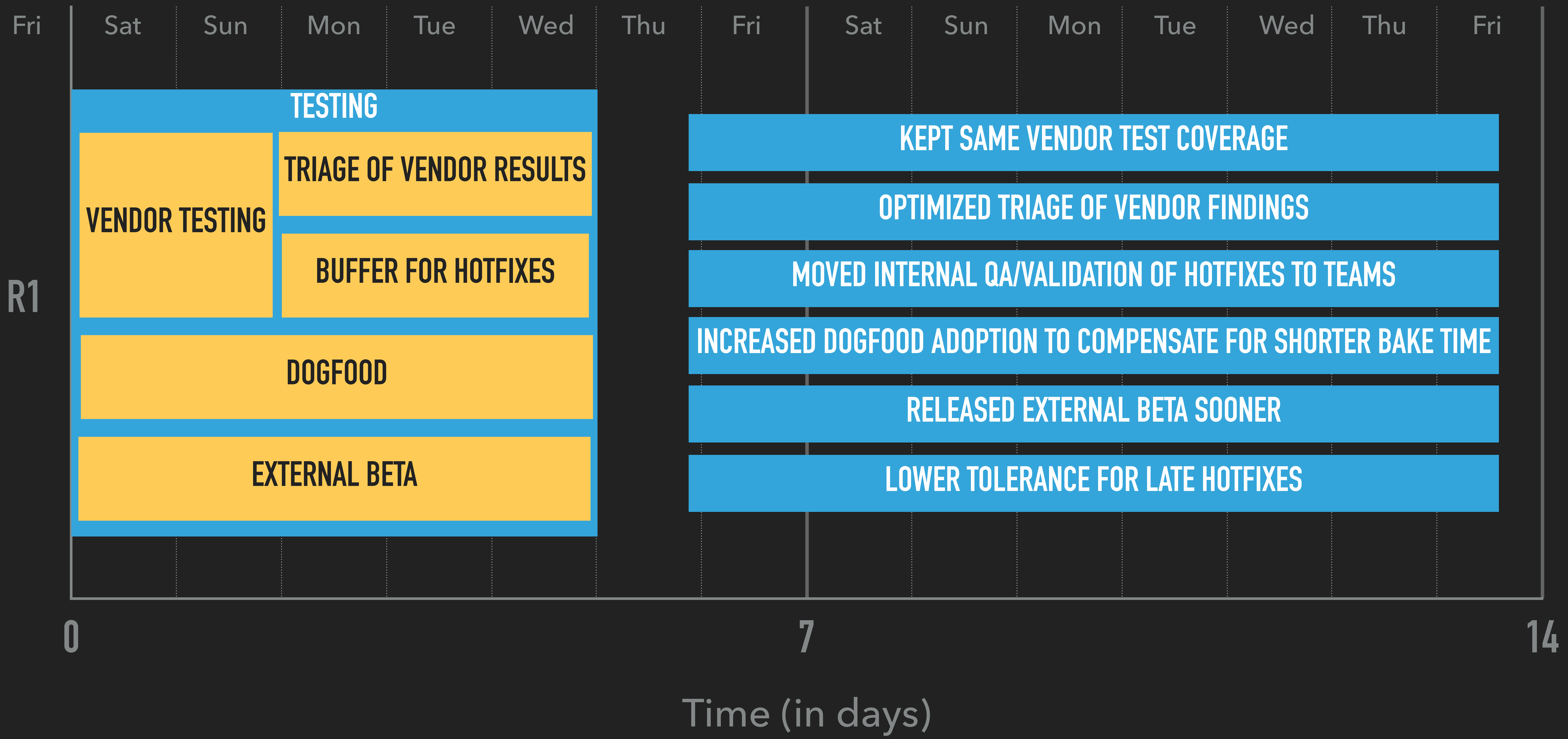
DIVING DEEPER


R7
R6
R5
R4
R3
R2
R1
0
7
14
21
28
35
42
49
56
63
70
77
84
91
Time (in days)

Slack Mobile Releases in 2022




Fri
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Sun
Mon
Tue
Wed
Thu
Fri
R1
TESTING
VENDOR TESTING
TRIAGE OF VENDOR RESULTS
INTERNAL QA/VALIDATION OF HOTFIXES
DOGFOOD
EXTERNAL BETA
0
7
14
Time (in days)



TESTING

VENDOR TESTING

TRIAGE OF VENDOR RESULTS

BUFFER FOR HOTFIXES

DOGFOOD

EXTERNAL BETA

KEPT SAME VENDOR TEST COVERAGE

OPTIMIZED TRIAGE OF VENDOR FINDINGS

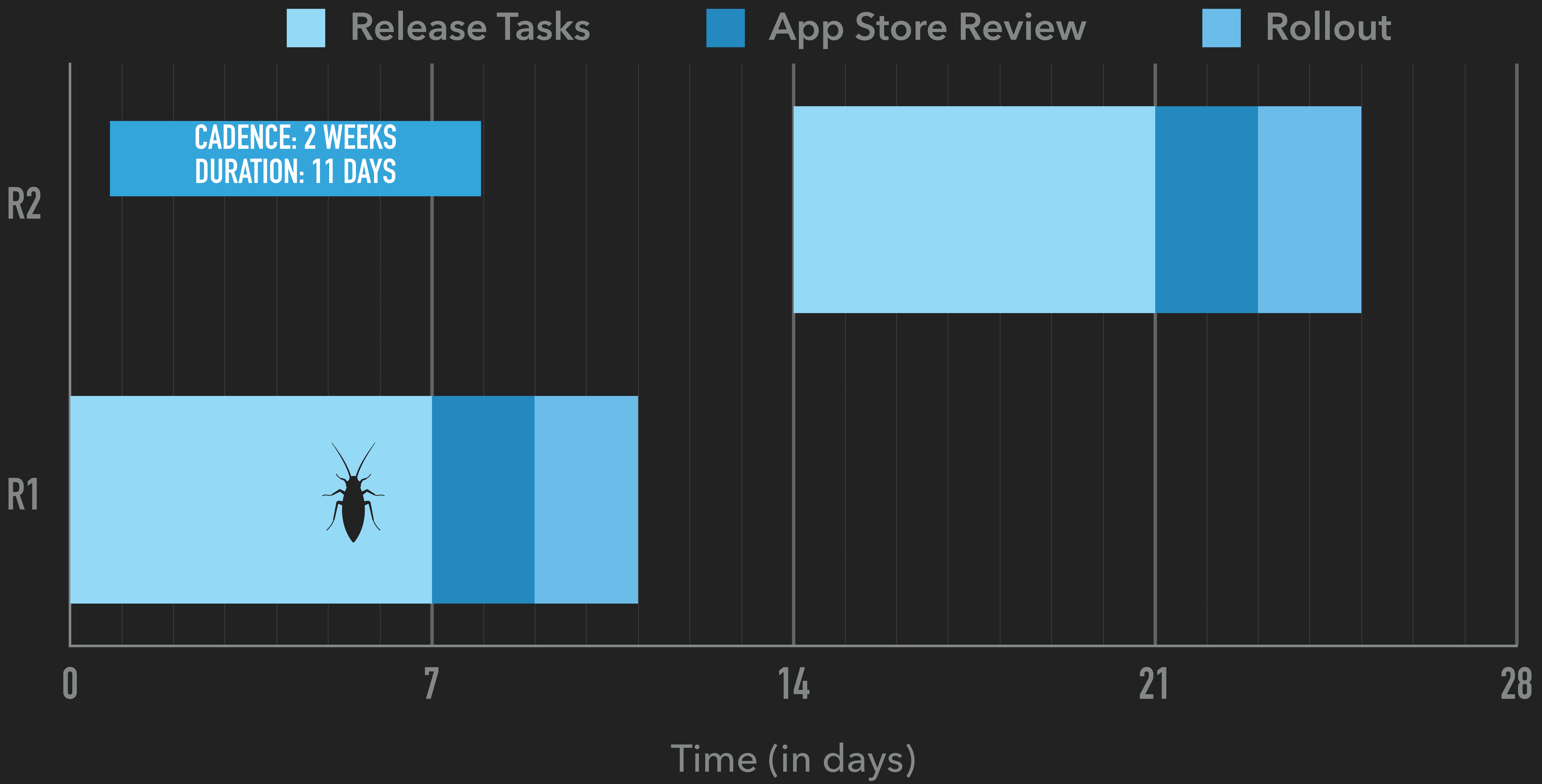
MOVED INTERNAL QA/VALIDATION OF HOTFIXES TO TEAMS

INCREASED DOGFOOD ADOPTION TO COMPENSATE FOR SHORTER BAKE TIME

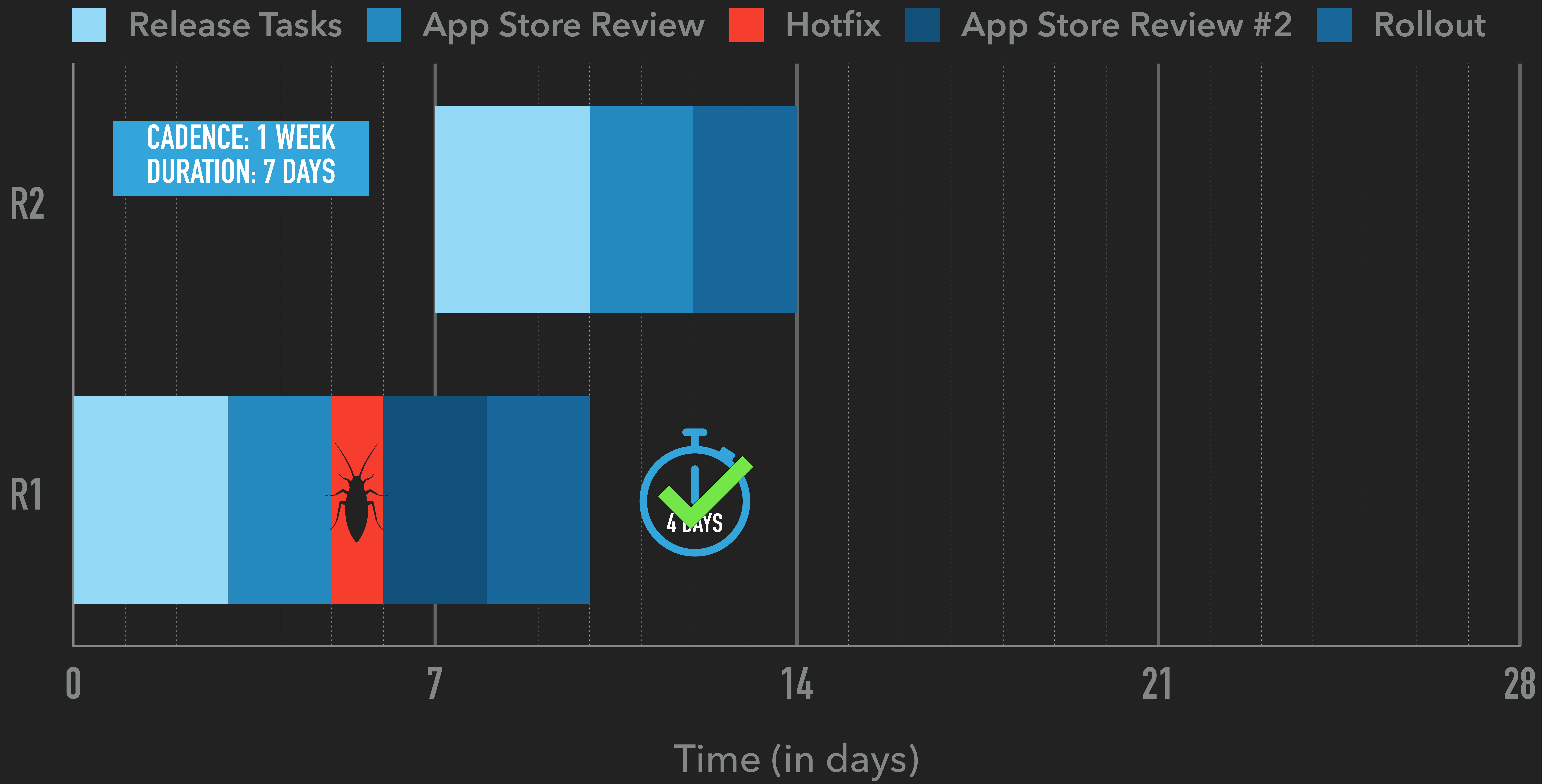
RELEASED EXTERNAL BETA SOONER

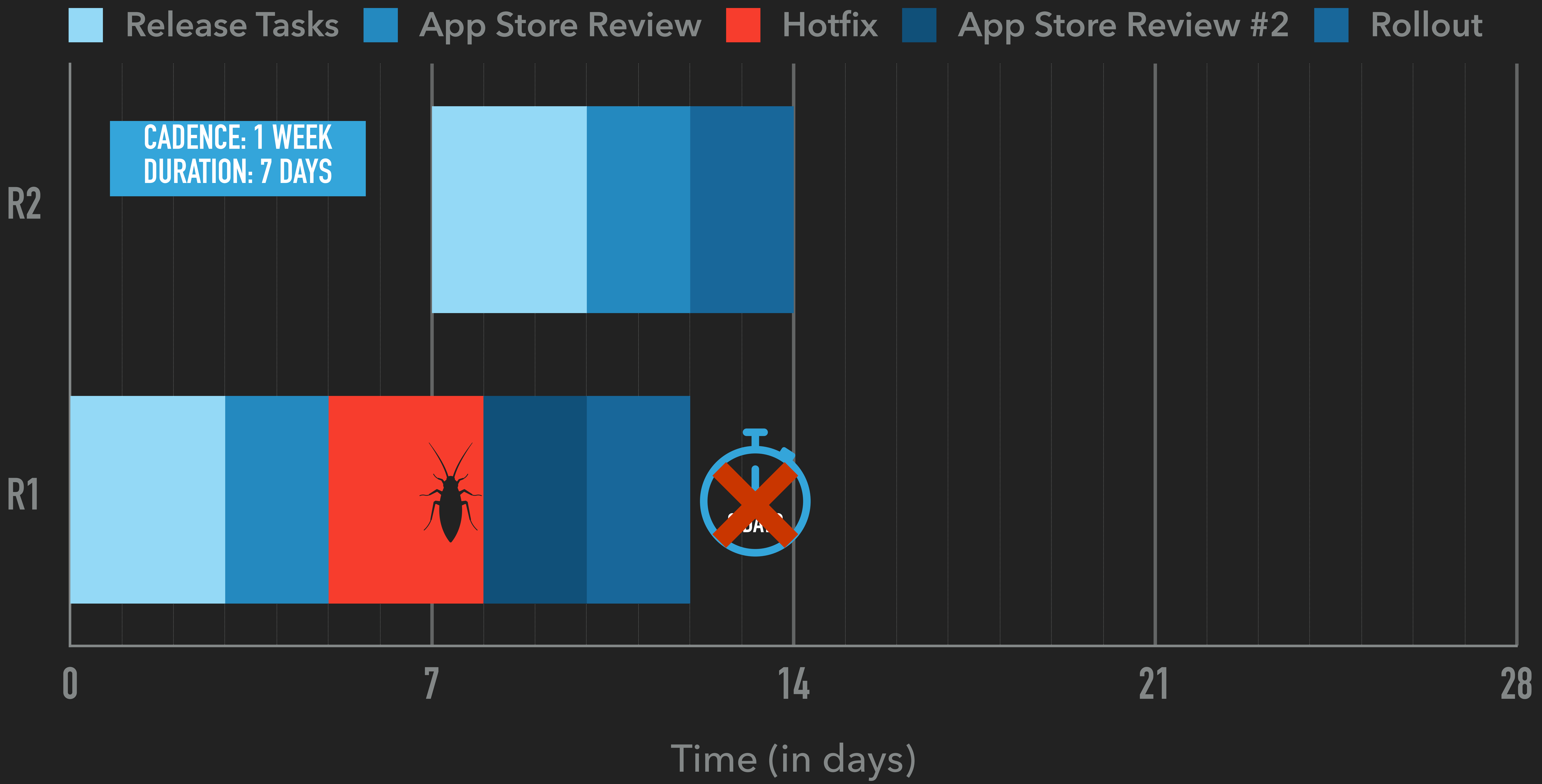
LOWER TOLERANCE FOR LATE HOTFIXES

HOTFIXES

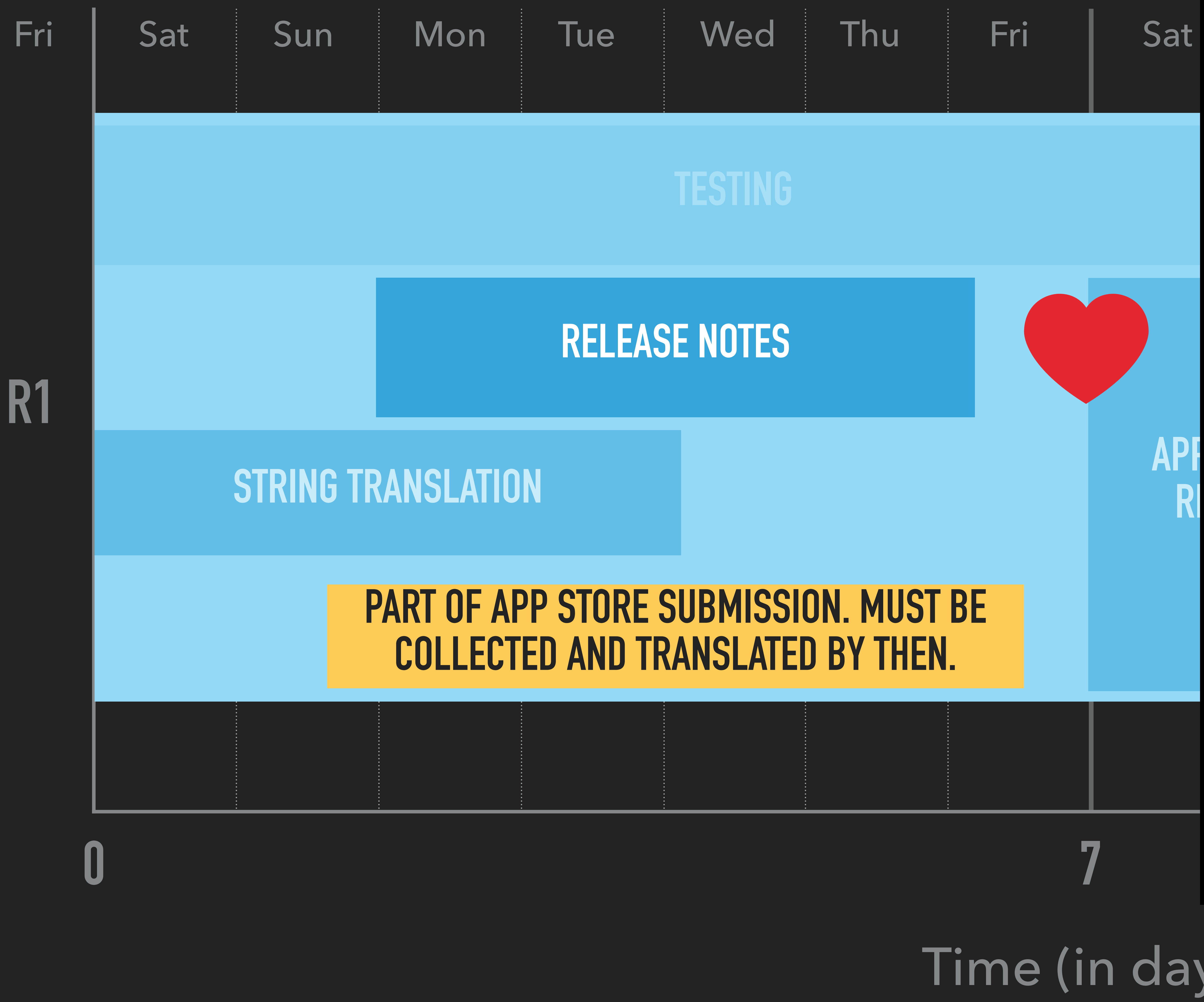


CADENCE: 2 WEEKS
DURATION: 11 DAYS

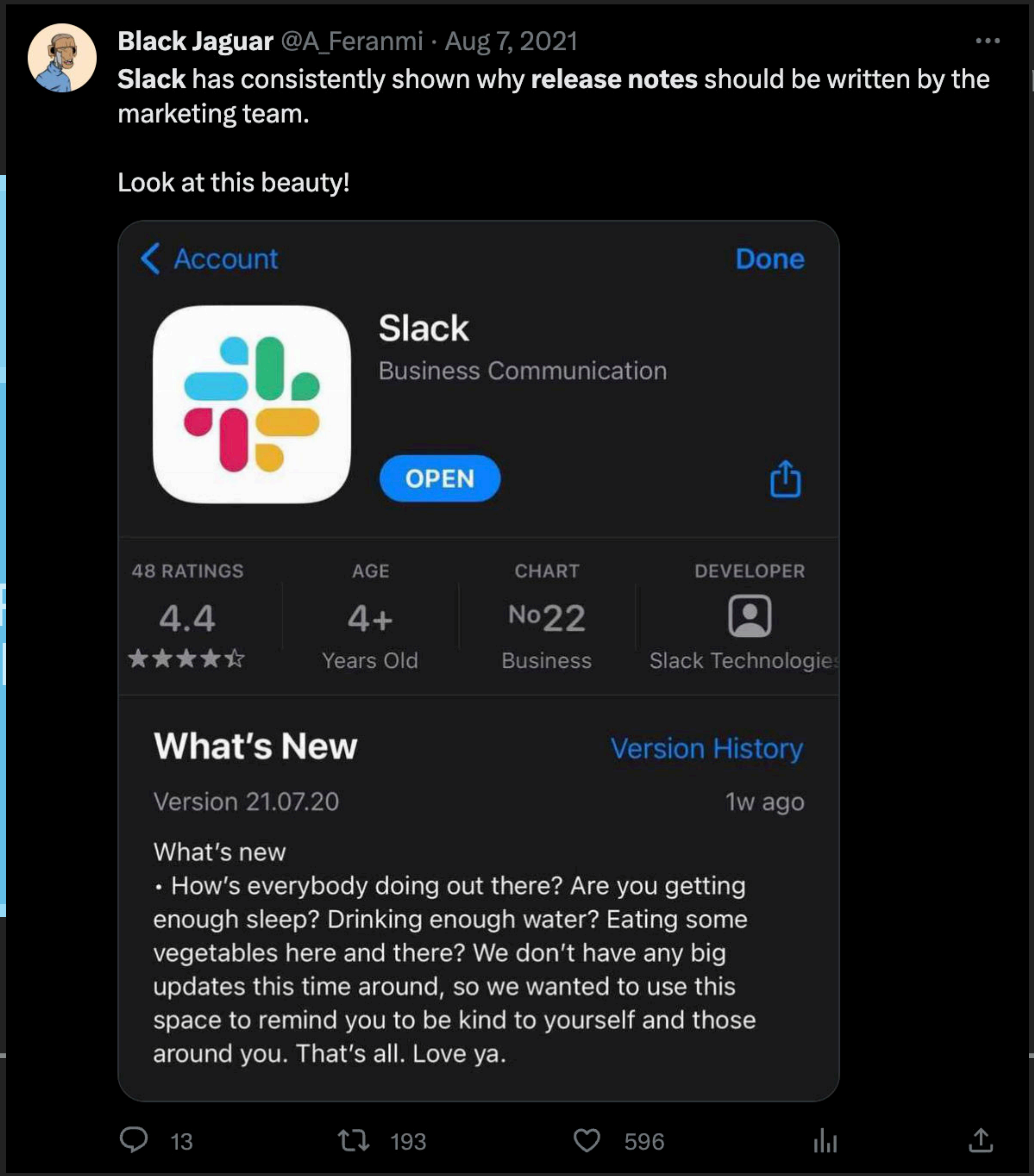

Release Tasks
App Store Review
Hotfix
App Store Review #2
Rollout
CADENCE: 1 WEEK
DURATION: 7 DAYS
R2
R1
4 DAYS
0
7
14
21
28
Time (in days)



CADENCE: 1 WEEK
DURATION: 7 DAYS



PART OF APP STORE SUBMISSION. MUST BE COLLECTED AND TRANSLATED BY THEN.



Black Jaguar @A_Feranmi · Aug 7, 2021

Slack has consistently shown why **release notes** should be written by the marketing team.

Look at this beauty!

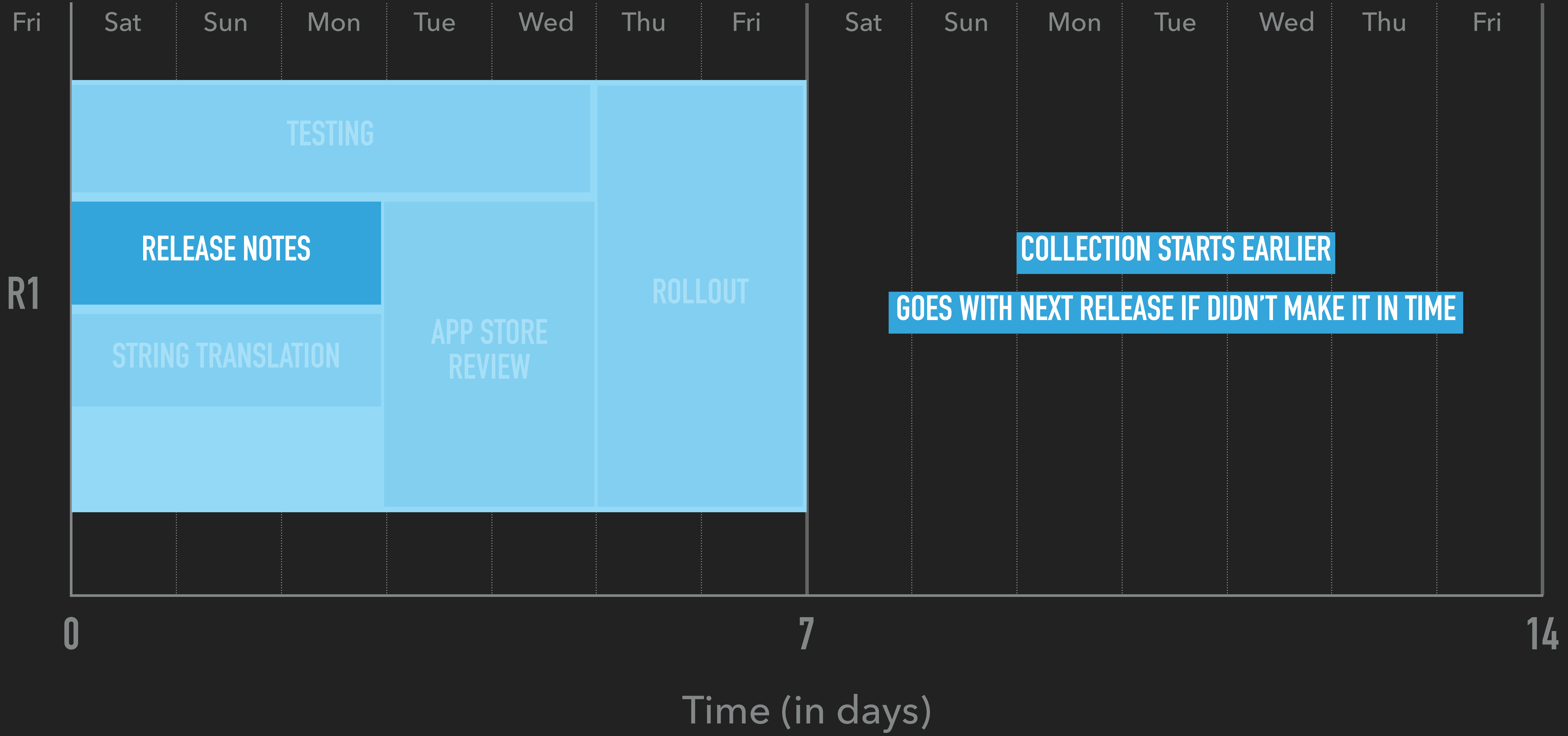
What's New

Version History

Version 21.07.20

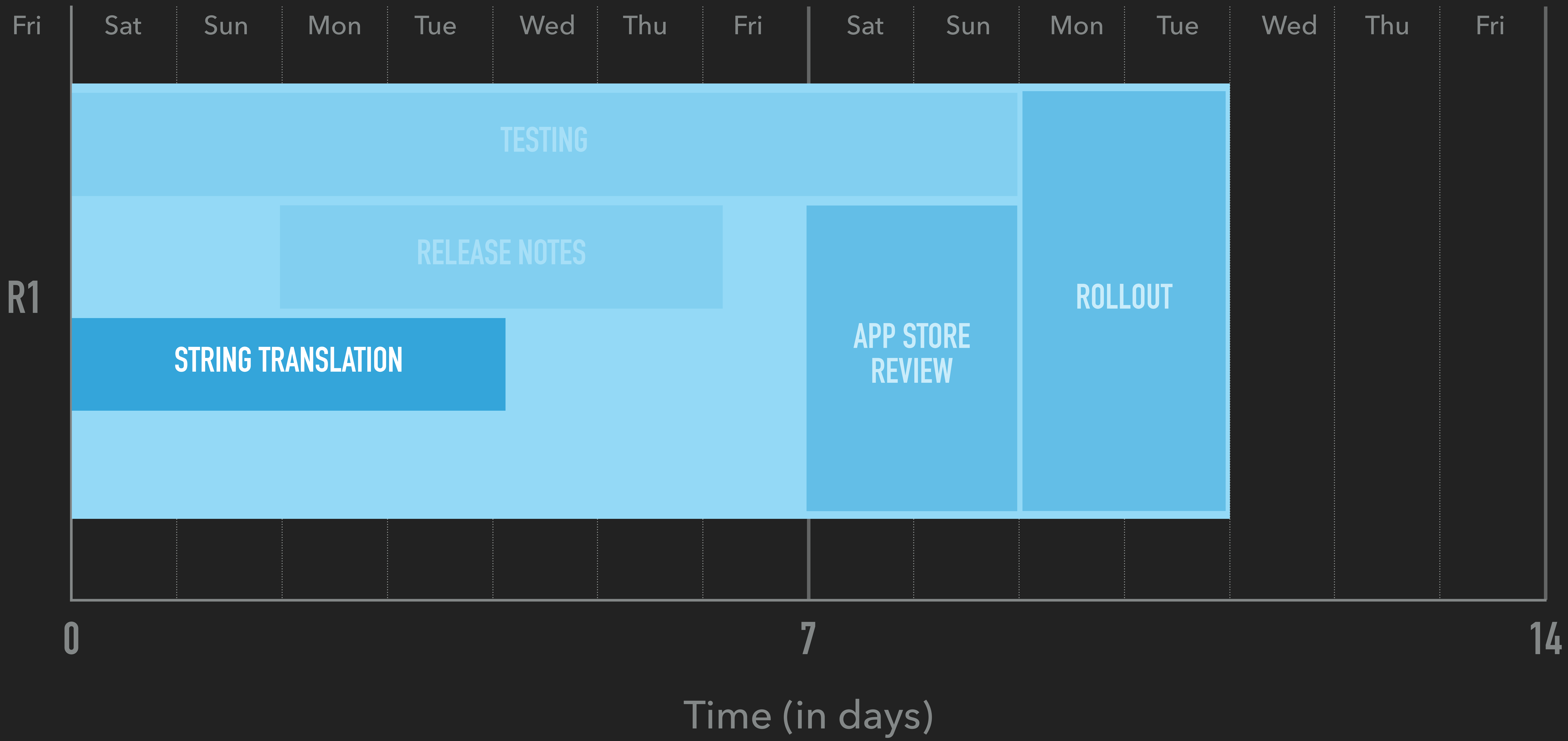
1w ago

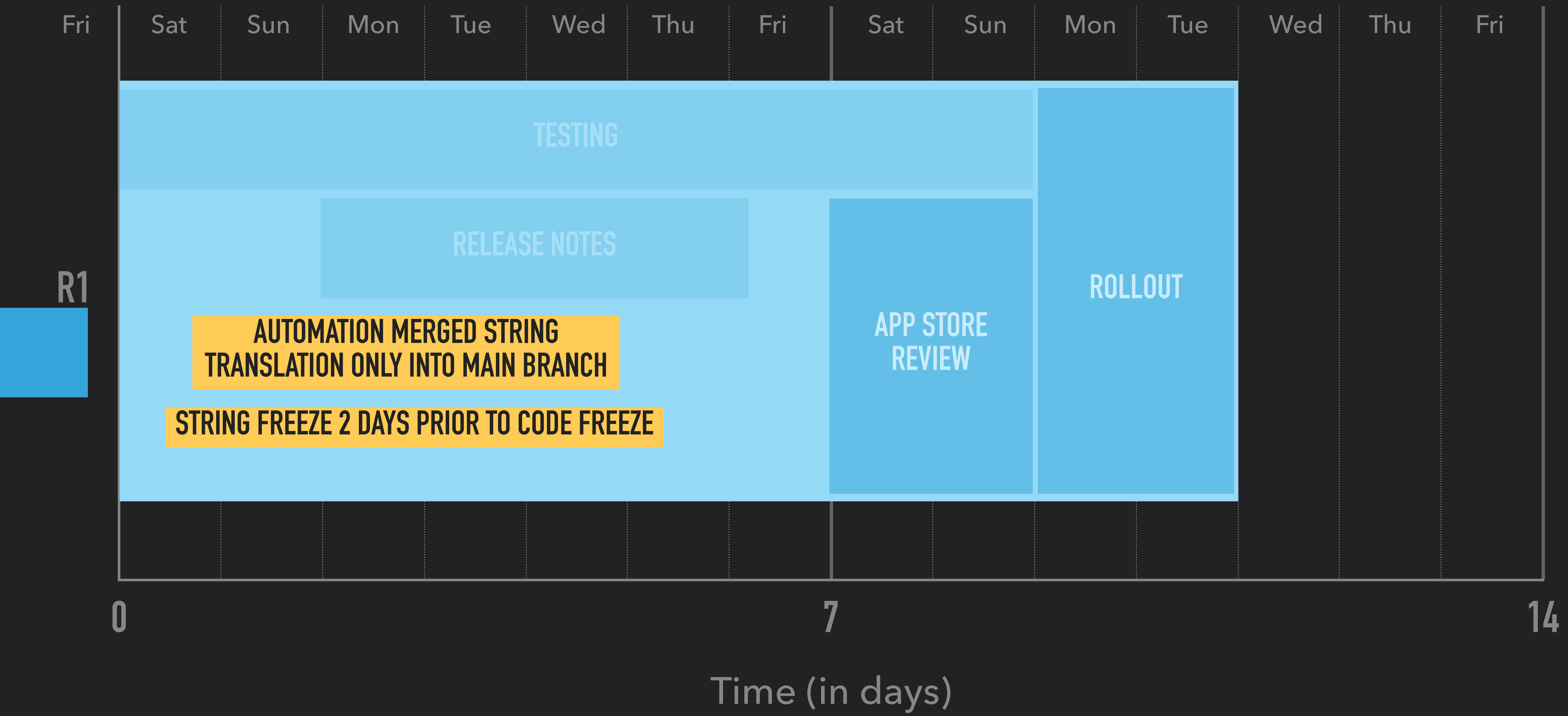
What's new
• How's everybody doing out there? Are you getting enough sleep? Drinking enough water? Eating some vegetables here and there? We don't have any big updates this time around, so we wanted to use this space to remind you to be kind to yourself and those around you. That's all. Love ya.



COLLECTION STARTS EARLIER

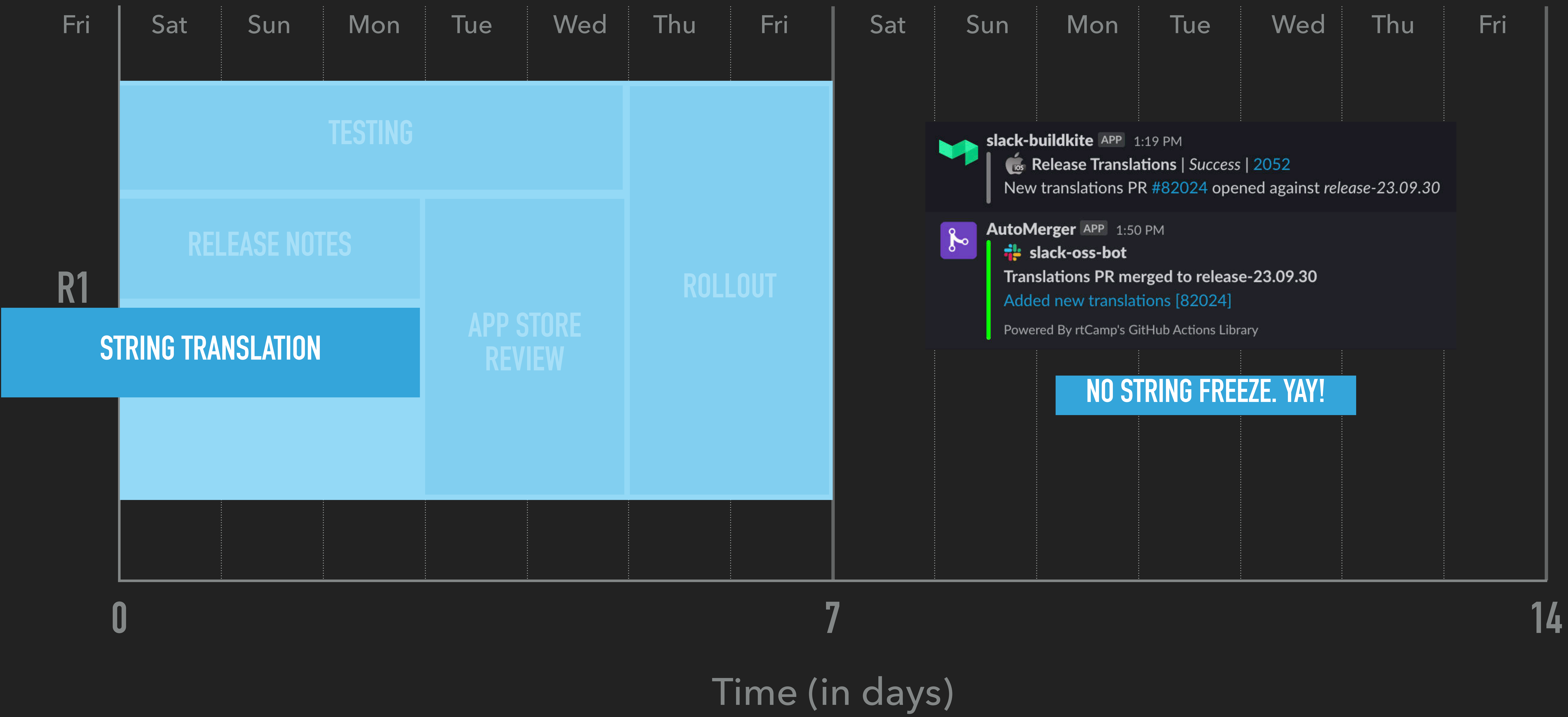
GOES WITH NEXT RELEASE IF DIDN'T MAKE IT IN TIME


Fri
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Sun
Mon
Tue
Wed
Thu
Fri
R1
TESTING
RELEASE NOTES
STRING TRANSLATION
APP STORE REVIEW
ROLLOUT
0
7
14
Time (in days)



AUTOMATION MERGED STRING TRANSLATION ONLY INTO MAIN BRANCH

STRING FREEZE 2 DAYS PRIOR TO CODE FREEZE



slack-buildkite APP 1:19 PM

Release Translations | *Success* | 2052

New translations PR #82024 opened against *release-23.09.30*

AutoMerger APP 1:50 PM

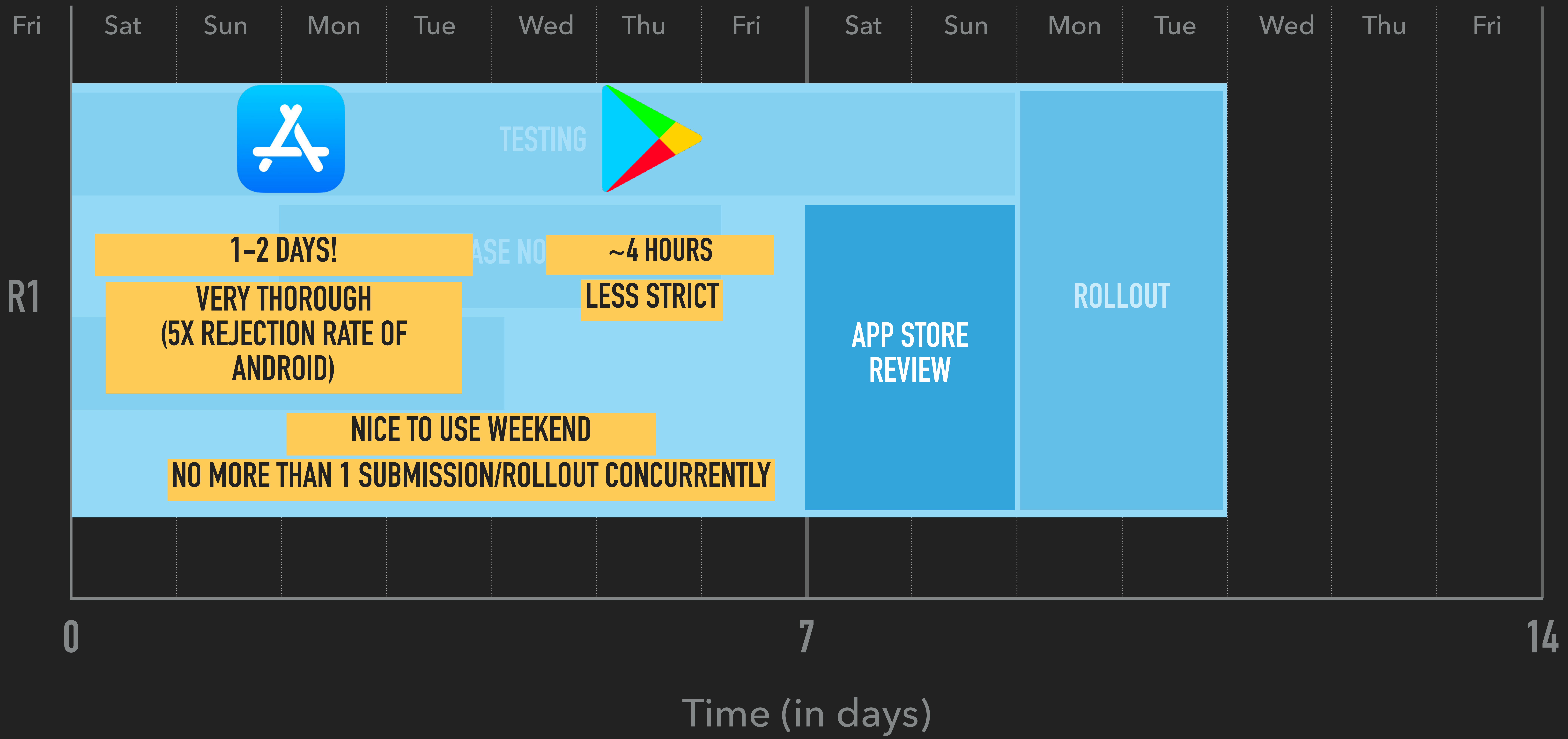
slack-oss-bot

Translations PR merged to release-23.09.30

Added new translations [82024]

Powered By rtCamp's GitHub Actions Library

NO STRING FREEZE. YAY!



1–2 DAYS!

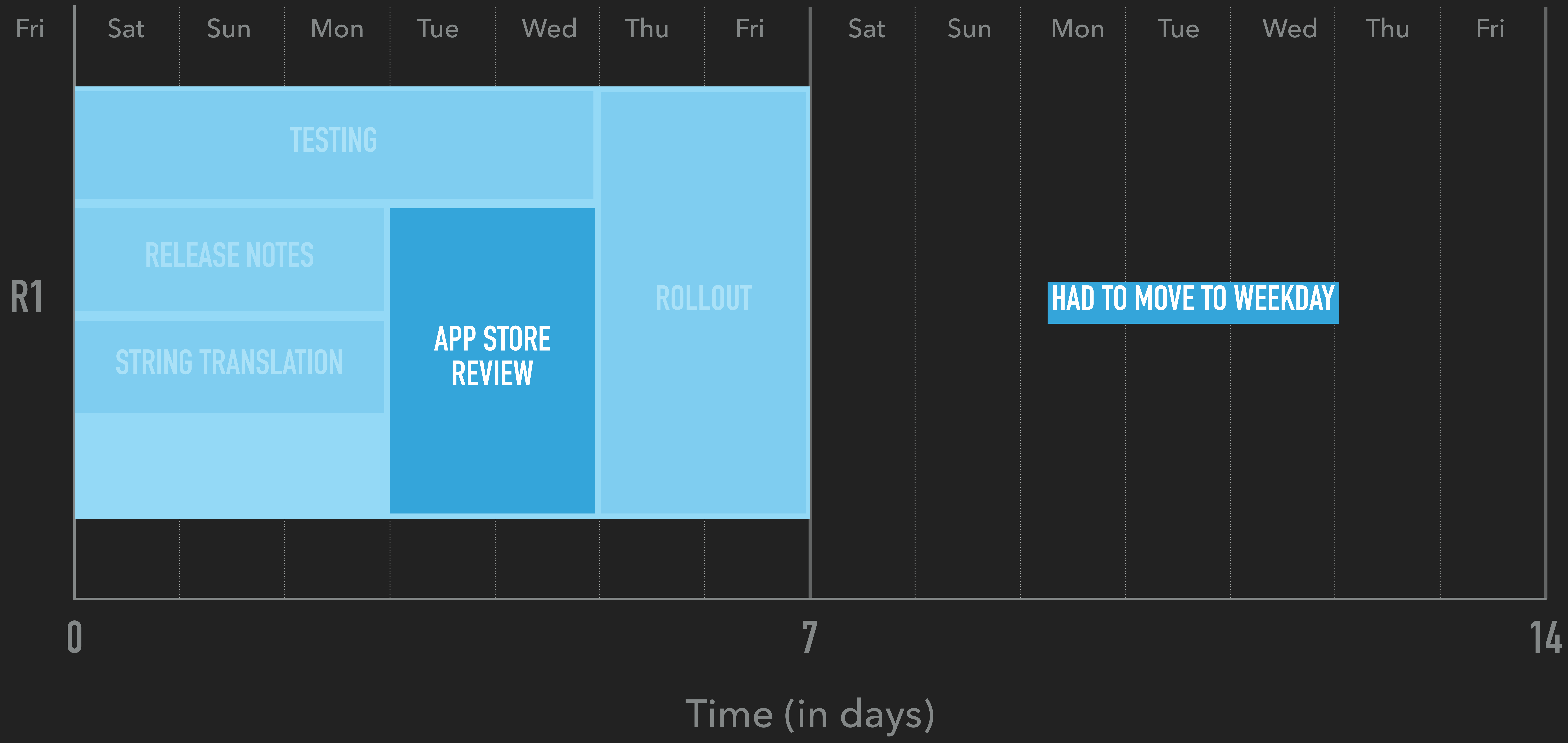
VERY THOROUGH
(5X REJECTION RATE OF
ANDROID)

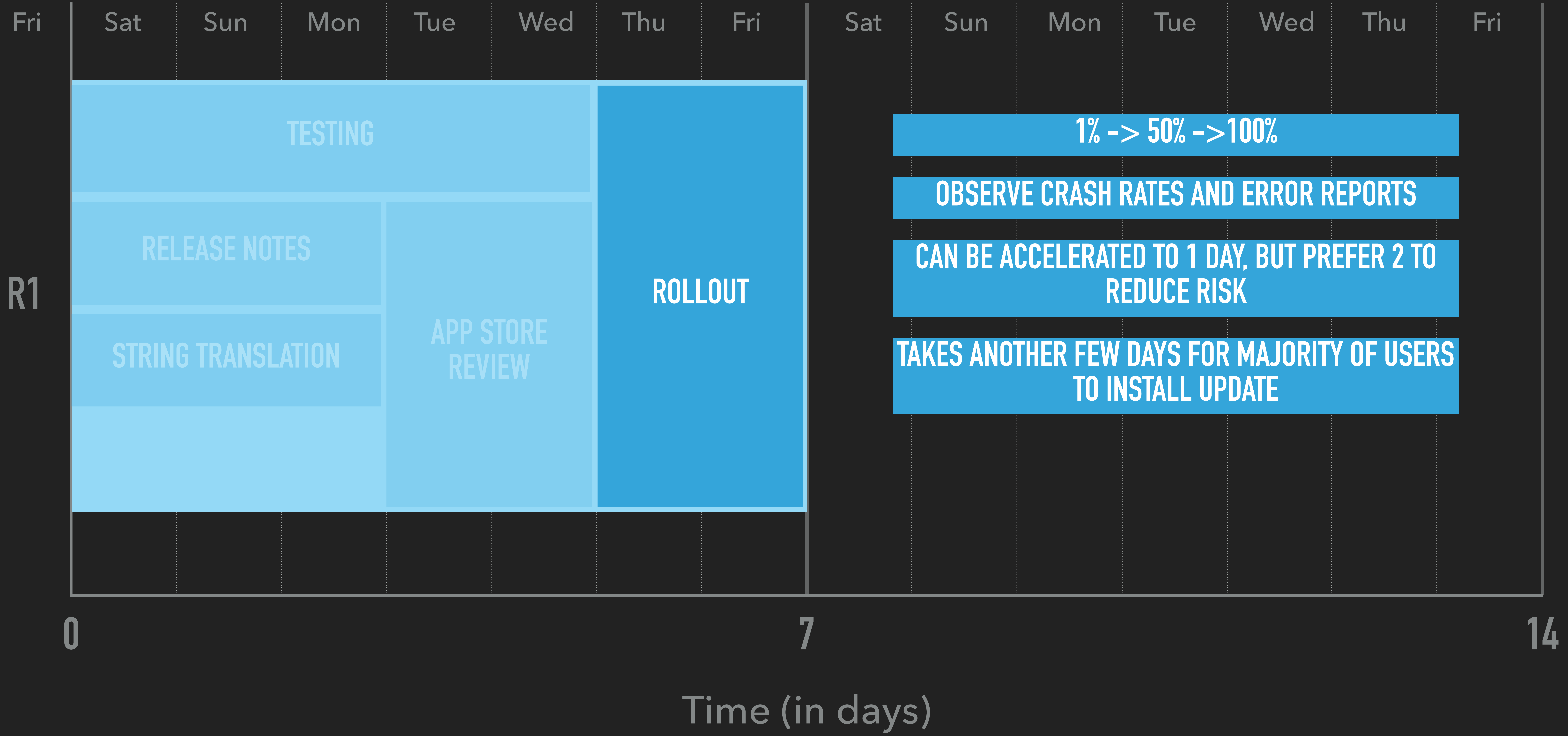
~4 HOURS

LESS STRICT

NICE TO USE WEEKEND

NO MORE THAN 1 SUBMISSION/ROLLOUT CONCURRENTLY


Fri
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Sun
Mon
Tue
Wed
Thu
Fri
R1
TESTING
RELEASE NOTES
STRING TRANSLATION
APP STORE REVIEW
ROLLOUT
HAD TO MOVE TO WEEKDAY
0
7
14
Time (in days)

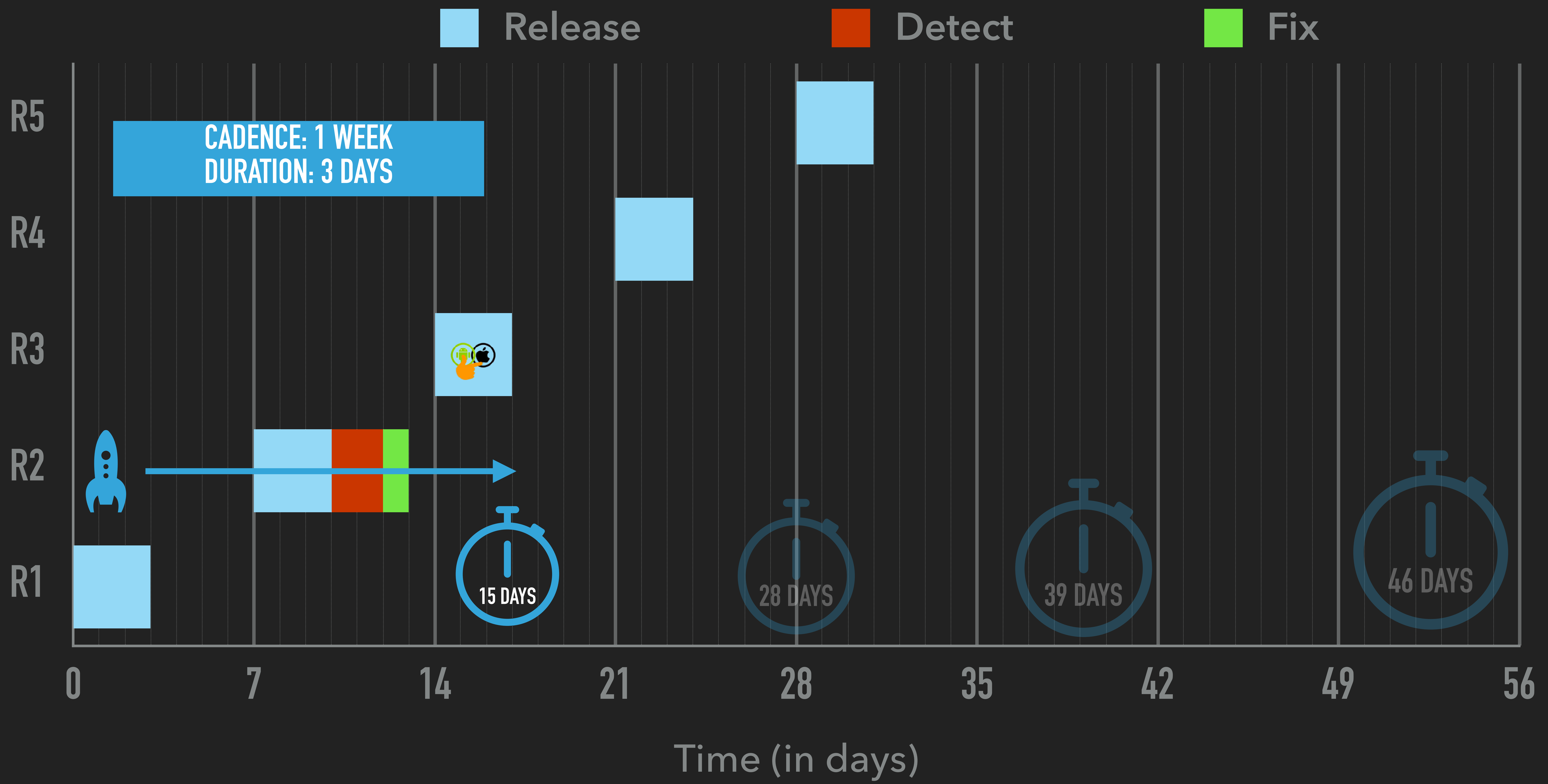


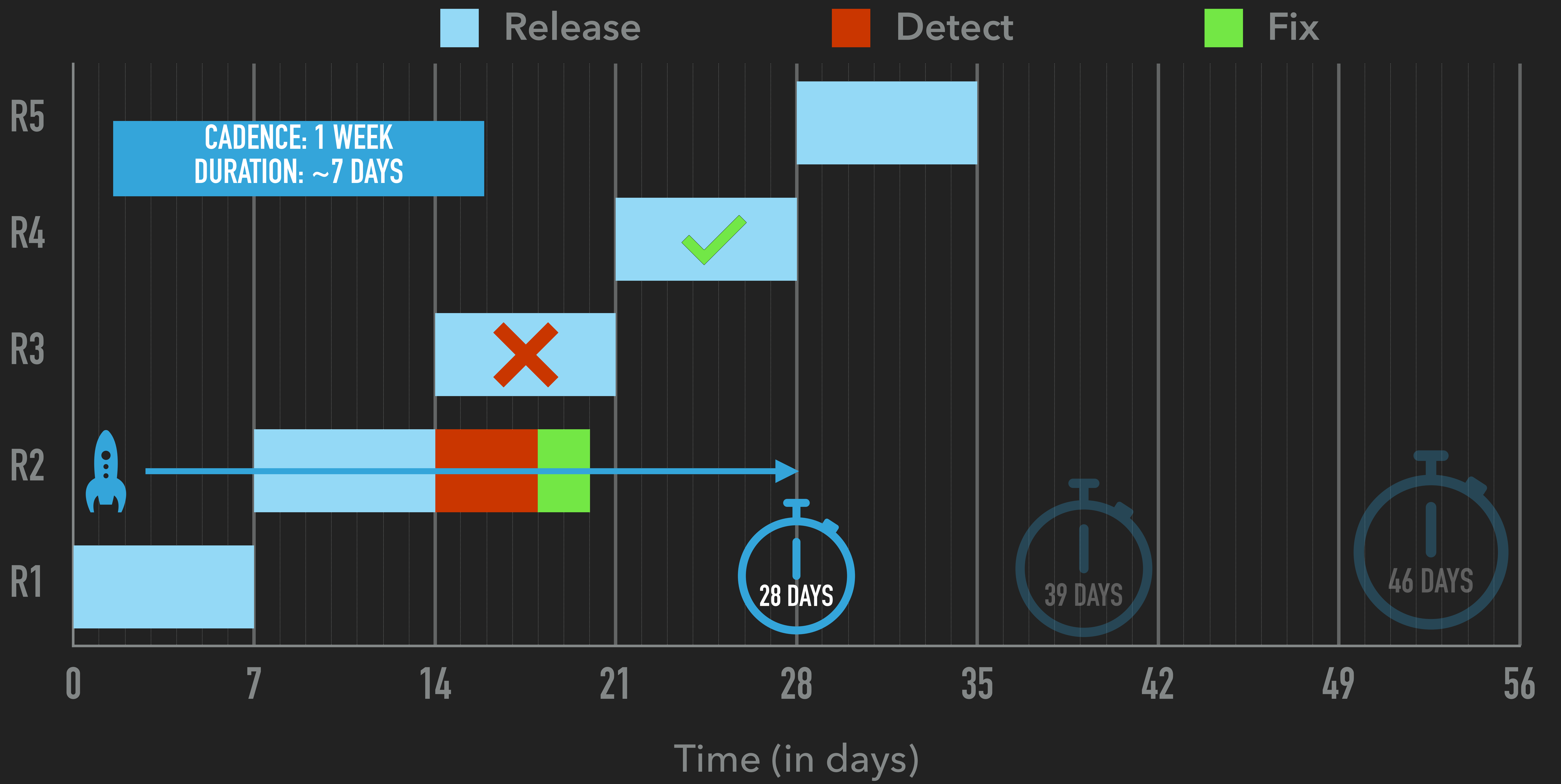
1% -> 50% ->100%

OBSERVE CRASH RATES AND ERROR REPORTS

CAN BE ACCELERATED TO 1 DAY, BUT PREFER 2 TO REDUCE RISK

TAKES ANOTHER FEW DAYS FOR MAJORITY OF USERS TO INSTALL UPDATE

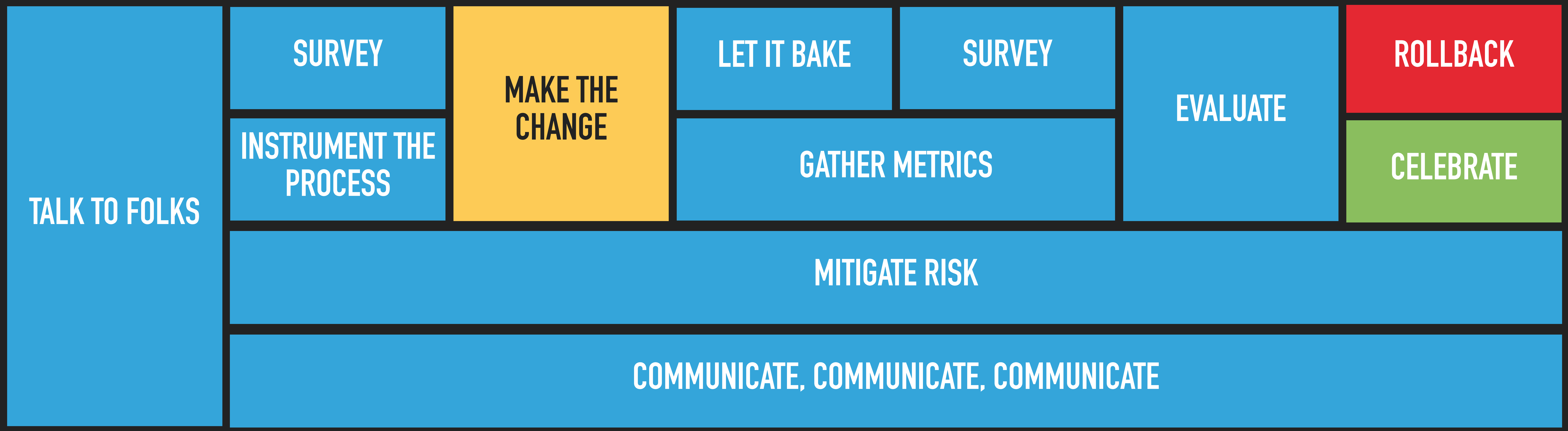

Release
Detect
Fix
R5
R4
R3
R2
R1
CADENCE: 1 WEEK
DURATION: 3 DAYS
15 DAYS
28 DAYS
39 DAYS
46 DAYS
0
7
14
21
28
35
42
49
56
Time (in days)



CADENCE: 1 WEEK
DURATION: ~7 DAYS

THE PROCESS OF CHANGE

THE PROCESS OF CHANGE



TALK TO FOLKS

SURVEY

INSTRUMENT THE PROCESS

MAKE THE CHANGE

LET IT BAKE

SURVEY

GATHER METRICS

EVALUATE

ROLLBACK

CELEBRATE

MITIGATE RISK

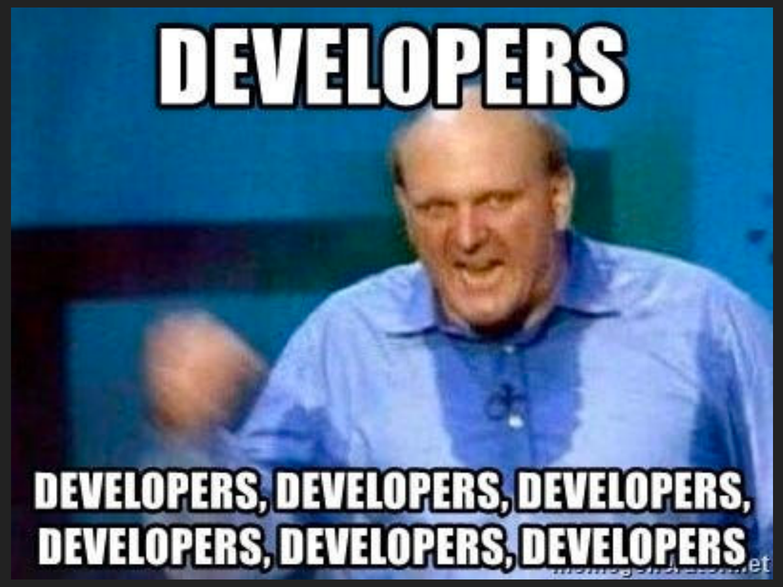
COMMUNICATE, COMMUNICATE, COMMUNICATE

THE PROCESS OF CHANGE

TALK TO FOLKS

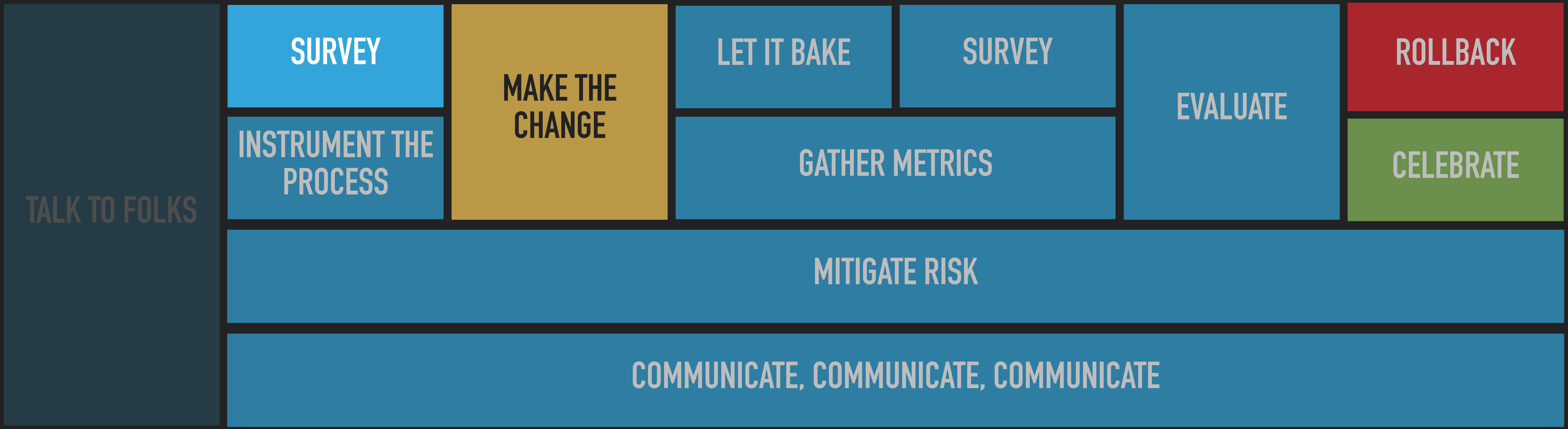
- Hopes
  - Faster iterations
  - More clarity on schedule
  - Reduced pressure to catch the train
  - Better stability
- Fears
  - Lower stability with reduced test time
  - Increased pressure to catch the train

- Not just developers


DEVELOPERS
DEVELOPERS, DEVELOPERS, DEVELOPERS,
DEVELOPERS, DEVELOPERS, DEVELOPERS

- Success Criteria
  - Increased velocity
  - Increased clarity
  - Reduced pressure
  - No regression in stability

THE PROCESS OF CHANGE



TALK TO FOLKS

SURVEY

INSTRUMENT THE PROCESS

MAKE THE CHANGE

LET IT BAKE

SURVEY

GATHER METRICS

EVALUATE

ROLLBACK

CELEBRATE

MITIGATE RISK

COMMUNICATE, COMMUNICATE, COMMUNICATE

SURVEY

- Based on success criteria from interviews
- Baseline for human perception
- Chance to
  - gather more info
  - let everyone be heard

THE PROCESS OF CHANGE

SURVEY

			Pressure	Schedule Clarity			Velocity			Safety	
			I/my team feel(s) pressure to "catch the release train" around string/code freeze.	I have clarity around when a change merged into the main branch will be available to dogfood users.	I have clarity around when a change merged into the main branch will be available to external beta users.	I have clarity around when a change merged into the main branch will become available to all external users.	I am satisfied with the speed at which changes from main branch become available to dogfood users.	I am satisfied with the speed at which changes from main branch become available to external beta users.	I am satisfied with the speed at which changes from main branch become available to all external users.	I am comfortable with releasing "green" builds from main branch to all external users without waiting for results of the manual regression test pass.	Feature flagging is an effective way to mitigate risk for changes in our app.
		Average	3.50	3.53	2.88	3.57	3.57	3.28	3.14	2.50	4.07
Scale	Average by discipline	**Development**	3.14	3.86	2.95	3.81	3.95	3.43	3.67	2.48	4.14
5 = Strongly Agree		**Design**	3.80	2.60	2.60	2.80	3.00	3.00	2.60	3.00	3.60
4 = Agree		**Engineering Management**	3.22	3.78	2.89	3.67	4.00	3.00	2.89	2.33	3.78
3 = Neutral		**Product Management**	4.00	2.79	2.36	3.36	2.79	2.93	2.64	2.79	4.36
2 = Disagree		**Quality Engineering**	3.71	4.00	3.29	3.29	3.71	3.86	3.29	2.14	4.14
1 = Strongly Disagree		**Program Management**	3.50	5.00	5.00	5.00	4.00	4.00	3.00	1.50	3.50
Legend	Average by Product/Infra	**Product Engineering**	3.81	3.60	2.95	3.67	3.55	3.36	3.21	2.55	4.24
bad		**Other**	3.33	2.67	2.50	2.50	3.00	3.00	2.33	2.67	3.33
green = good		**Infrastructure**	2.30	3.80	2.80	3.80	4.00	3.10	3.30	2.20	3.80
	Average by Platform	**Android**	3.25	3.63	3.25	3.94	4.00	3.69	3.69	2.38	4.25
		Android|iOS	3.89	3.33	2.81	3.56	3.22	3.07	2.74	2.52	4.07
		iOS	3.07	3.80	2.60	3.20	3.73	3.20	3.27	2.60	3.87

THE PROCESS OF CHANGE

SURVEY

			Pressure	Schedule Clarity			Velocity			Safety	
			I/my team feel(s) pressure to "catch the release train" around string/code freeze.	I have clarity around when a change merged into the main branch will be available to dogfood users.	I have clarity around when a change merged into the main branch will be available to external beta users.	I have clarity around when a change merged into the main branch will become available to all external users.	I am satisfied with the speed at which changes from main branch become available to dogfood users.	I am satisfied with the speed at which changes from main branch become available to external beta users.	I am satisfied with the speed at which changes from main branch become available to all external users.	I am comfortable with releasing "green" builds from main branch to all external users without waiting for results of the manual regression test pass.	Feature flagging is an effective way to mitigate risk for changes in our app.
		Average	3.50	3.53	2.88	3.57	3.57	3.28	3.14	2.50	4.07
Scale	Average by discipline	**Development**	3.14	3.86	2.95	3.81	3.95	3.43	3.67	2.48	4.14
5 = Strongly Agree		**Design**	3.80	2.60	2.60	2.80	3.00	3.00	2.60	3.00	3.60
4 = Agree		**Engineering Management**	3.22	3.78	2.89	3.67	4.00	3.00	2.89	2.33	3.78
3 = Neutral		**Product Management**	4.00	2.79	2.36	3.36	2.79	2.93	2.64	2.79	4.36
2 = Disagree		**Quality Engineering**	3.71	4.00	3.29	3.29	3.71	3.86	3.29	2.14	4.14
1 = Strongly Disagree		**Program Management**	3.50	5.00	5.00	5.00	4.00	4.00	3.00	1.50	3.50
Legend	Average by Product/Infra	**Product Engineering**	3.81	3.60	2.95	3.67	3.55	3.36	3.21	2.55	4.24
bad		**Other**	3.33	2.67	2.50	2.50	3.00	3.00	2.33	2.67	3.33
green = good		**Infrastructure**	2.30	3.80	2.80	3.80	4.00	3.10	3.30	2.20	3.80
	Average by Platform	**Android**	3.25	3.63	3.25	3.94	4.00	3.69	3.69	2.38	4.25
		Android|iOS	3.89	3.33	2.81	3.56	3.22	3.07	2.74	2.52	4.07
		iOS	3.07	3.80	2.60	3.20	3.73	3.20	3.27	2.60	3.87

ON PRESSURE TO CATCH THE RELEASE TRAIN AND SPEED

SURVEY

> “WE TRY AND SQUEEZE UPDATES AND IMPROVEMENTS IN THE UPCOMING RELEASE, BECAUSE THE NEXT ONE IS TOO FAR. BY THE TIME THE CUSTOMER SEES OUR WORK – IT'S 1 MONTH OUT. IT SLOWS DOWN PACE OF SHIPPING, EXPERIMENTATION AND LEARNING ON MOBILE PLATFORMS. DESKTOP IS ABLE TO SHIP TO GA IN A DAY (WHICH WOULD BE THE DREAM). BUT, I CAN ONLY HOPE WE CAN MAKE IT QUICKER THAN WHAT IT IS TODAY.”

Designer at Slack

THE PROCESS OF CHANGE

SURVEY

Scale
5 = Strongly Agree
4 = Agree
3 = Neutral
2 = Disagree
1 = Strongly Disagree

Legend
🟥 bad
🟩 green = good

		Pressure	Schedule Clarity			Velocity			Safety	
		I/my team feel(s) pressure to "catch the release train" around string/code freeze.	I have clarity around when a change merged into the main branch will be available to dogfood users.	I have clarity around when a change merged into the main branch will be available to external beta users.	I have clarity around when a change merged into the main branch will become available to all external users.	I am satisfied with the speed at which changes from main branch become available to dogfood users.	I am satisfied with the speed at which changes from main branch become available to external beta users.	I am satisfied with the speed at which changes from main branch become available to all external users.	I am comfortable with releasing "green" builds from main branch to all external users without waiting for results of the manual regression test pass.	Feature flagging is an effective way to mitigate risk for changes in our app.
	Average	3.50	3.53	2.88	3.57	3.57	3.28	3.14	2.50	4.07
Average by discipline	Development	3.14	3.86	2.95	3.81	3.95	3.43	3.67	2.48	4.14
	Design	3.80	2.60	2.60	2.80	3.00	3.00	2.60	3.00	3.60
	Engineering Management	3.22	3.78	2.89	3.67	4.00	3.00	2.89	2.33	3.78
	Product Management	4.00	2.79	2.36	3.36	2.79	2.93	2.64	2.79	4.36
	Quality Engineering	3.71	4.00	3.29	3.29	3.71	3.86	3.29	2.14	4.14
	Program Management	3.50	5.00	5.00	5.00	4.00	4.00	3.00	1.50	3.50
Average by Product/Infra	Product Engineering	3.81	3.60	2.95	3.67	3.55	3.36	3.21	2.55	4.24
	Other	3.33	2.67	2.50	2.50	3.00	3.00	2.33	2.67	3.33
	Infrastructure	2.30	3.80	2.80	3.80	4.00	3.10	3.30	2.20	3.80
Average by Platform	Android	3.25	3.63	3.25	3.94	4.00	3.69	3.69	2.38	4.25
	Android|iOS	3.89	3.33	2.81	3.56	3.22	3.07	2.74	2.52	4.07
	iOS	3.07	3.80	2.60	3.20	3.73	3.20	3.27	2.60	3.87

ON CLARITY AROUND SCHEDULE

SURVEY

"I HAVE NO IDEA — WHAT ARE EXTERNAL BETA USERS?"

A PRODUCT MANAGER

"THERE ARE SEVERAL CUT OFFS THAT I HAVE TO CONSISTENTLY ASK AGAIN OR SECOND GUESS JUST IN CASE I CONFUSE MYSELF OR CONFUSE OTHERS."

A DEVELOPER

THE PROCESS OF CHANGE

SURVEY

Pressure Schedule Clarity Velocity Safety

			I/my team feel(s) pressure to "catch the release train" around string/code freeze.	I have clarity around when a change merged into the main branch will be available to dogfood users.	I have clarity around when a change merged into the main branch will be available to external beta users.	I have clarity around when a change merged into the main branch will become available to all external users.	I am satisfied with the speed at which changes from main branch become available to dogfood users.	I am satisfied with the speed at which changes from main branch become available to external beta users.	I am satisfied with the speed at which changes from main branch become available to all external users.	I am comfortable with releasing "green" builds from main branch to all external users without waiting for results of the manual regression test pass.	Feature flagging is an effective way to mitigate risk for changes in our app.
		Average	3.50	3.53	2.88	3.57	3.57	3.28	3.14	2.50	4.07
Scale	Average by discipline	Development	3.14	3.86	2.95	3.81	3.95	3.43	3.67	2.48	4.14
5 = Strongly Agree		Design	3.80	2.60	2.60	2.80	3.00	3.00	2.60	3.00	3.60
4 = Agree		Engineering Management	3.22	3.78	2.89	3.67	4.00	3.00	2.89	2.33	3.78
3 = Neutral		Product Management	4.00	2.79	2.36	3.36	2.79	2.93	2.64	2.79	4.36
2 = Disagree		Quality Engineering	3.71	4.00	3.29	3.29	3.71	3.86	3.29	2.14	4.14
1 = Strongly Disagree		Program Management	3.50	5.00	5.00	5.00	4.00	4.00	3.00	1.50	3.50
Legend	Average by Product/Infra	Product Engineering	3.81	3.60	2.95	3.67	3.55	3.36	3.21	2.55	4.24
🟥 bad		Other	3.33	2.67	2.50	2.50	3.00	3.00	2.33	2.67	3.33
🟩 green = good		Infrastructure	2.30	3.80	2.80	3.80	4.00	3.10	3.30	2.20	3.80
	Average by Platform	Android	3.25	3.63	3.25	3.94	4.00	3.69	3.69	2.38	4.25
		Android|iOS	3.89	3.33	2.81	3.56	3.22	3.07	2.74	2.52	4.07
		iOS	3.07	3.80	2.60	3.20	3.73	3.20	3.27	2.60	3.87

ON RELEASE SPEED

SURVEY

“OTHER PLACES I'VE BEEN THE WAIT HAS BEEN EVEN LONGER. I FIND SLACK'S CURRENT TIMING FINE.”

A DEVELOPER

“2 WEEKS IS A LONG TIME IN THE MODERN WORLD OF SOFTWARE”

A DESIGNER

THE PROCESS OF CHANGE

SURVEY

			Pressure	Schedule Clarity			Velocity			Safety	
			I/my team feel(s) pressure to "catch the release train" around string/code freeze.	I have clarity around when a change merged into the main branch will be available to dogfood users.	I have clarity around when a change merged into the main branch will be available to external beta users.	I have clarity around when a change merged into the main branch will become available to all external users.	I am satisfied with the speed at which changes from main branch become available to dogfood users.	I am satisfied with the speed at which changes from main branch become available to external beta users.	I am satisfied with the speed at which changes from main branch become available to all external users.	I am comfortable with releasing "green" builds from main branch to all external users without waiting for results of the manual regression test pass.	Feature flagging is an effective way to mitigate risk for changes in our app.
		Average	3.50	3.53	2.88	3.57	3.57	3.28	3.14	2.50	4.07
Scale	Average by discipline	Development	3.14	3.86	2.95	3.81	3.95	3.43	3.67	2.48	4.14
5 = Strongly Agree		Design	3.80	2.60	2.60	2.80	3.00	3.00	2.60	3.00	3.60
4 = Agree		Engineering Management	3.22	3.78	2.89	3.67	4.00	3.00	2.89	2.33	3.78
3 = Neutral		Product Management	4.00	2.79	2.36	3.36	2.79	2.93	2.64	2.79	4.36
2 = Disagree		Quality Engineering	3.71	4.00	3.29	3.29	3.71	3.86	3.29	2.14	4.14
1 = Strongly Disagree		Program Management	3.50	5.00	5.00	5.00	4.00	4.00	3.00	1.50	3.50
Legend	Average by Product/Infra	Product Engineering	3.81	3.60	2.95	3.67	3.55	3.36	3.21	2.55	4.24
bad		Other	3.33	2.67	2.50	2.50	3.00	3.00	2.33	2.67	3.33
green = good		Infrastructure	2.30	3.80	2.80	3.80	4.00	3.10	3.30	2.20	3.80
	Average by Platform	Android	3.25	3.63	3.25	3.94	4.00	3.69	3.69	2.38	4.25
		Android|iOS	3.89	3.33	2.81	3.56	3.22	3.07	2.74	2.52	4.07
		iOS	3.07	3.80	2.60	3.20	3.73	3.20	3.27	2.60	3.87

ON RELEASING WITHOUT MANUAL QA

SURVEY

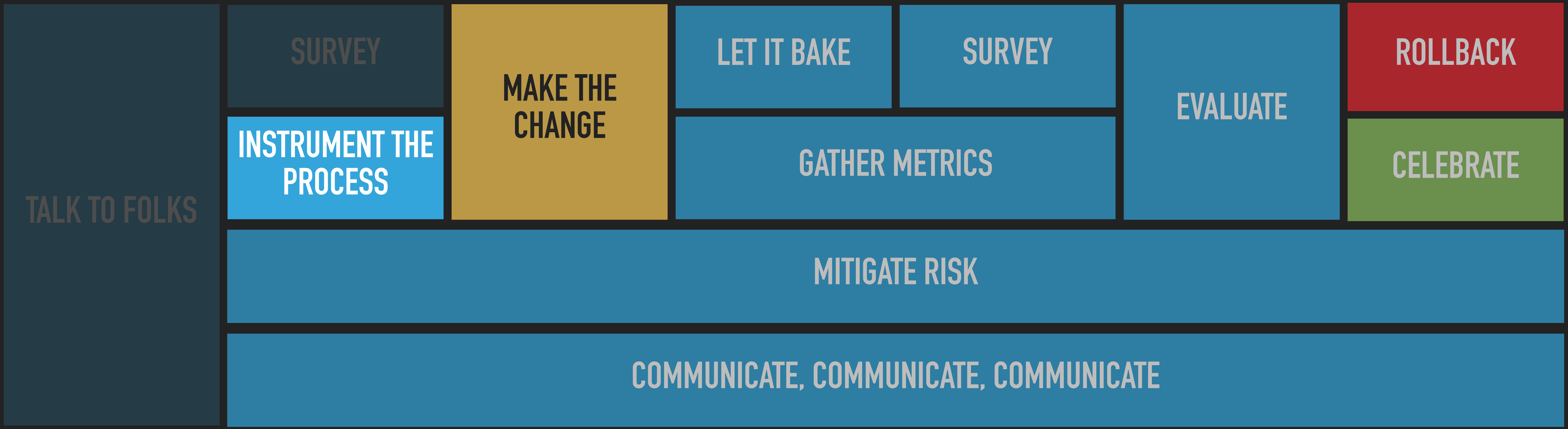
“WE NEED TO IMPROVE (A LOT) ON OUR AUTOMATE TESTING BEFORE I CAN COMFORTABLE WITH THIS.”

A DEVELOPER

“JUST BECAUSE THE BUILD IS "GREEN", IT DOESN'T MEAN THAT THERE ARE NO CRITICAL ISSUES. WE SHOULD ALWAYS HAVE MANUAL REGRESSION PASS BEFORE RELEASING TO EXTERNAL USERS”

AN ENGINEERING MANAGER

THE PROCESS OF CHANGE



TALK TO FOLKS

SURVEY

INSTRUMENT THE PROCESS

MAKE THE CHANGE

LET IT BAKE

SURVEY

GATHER METRICS

EVALUATE

ROLLBACK

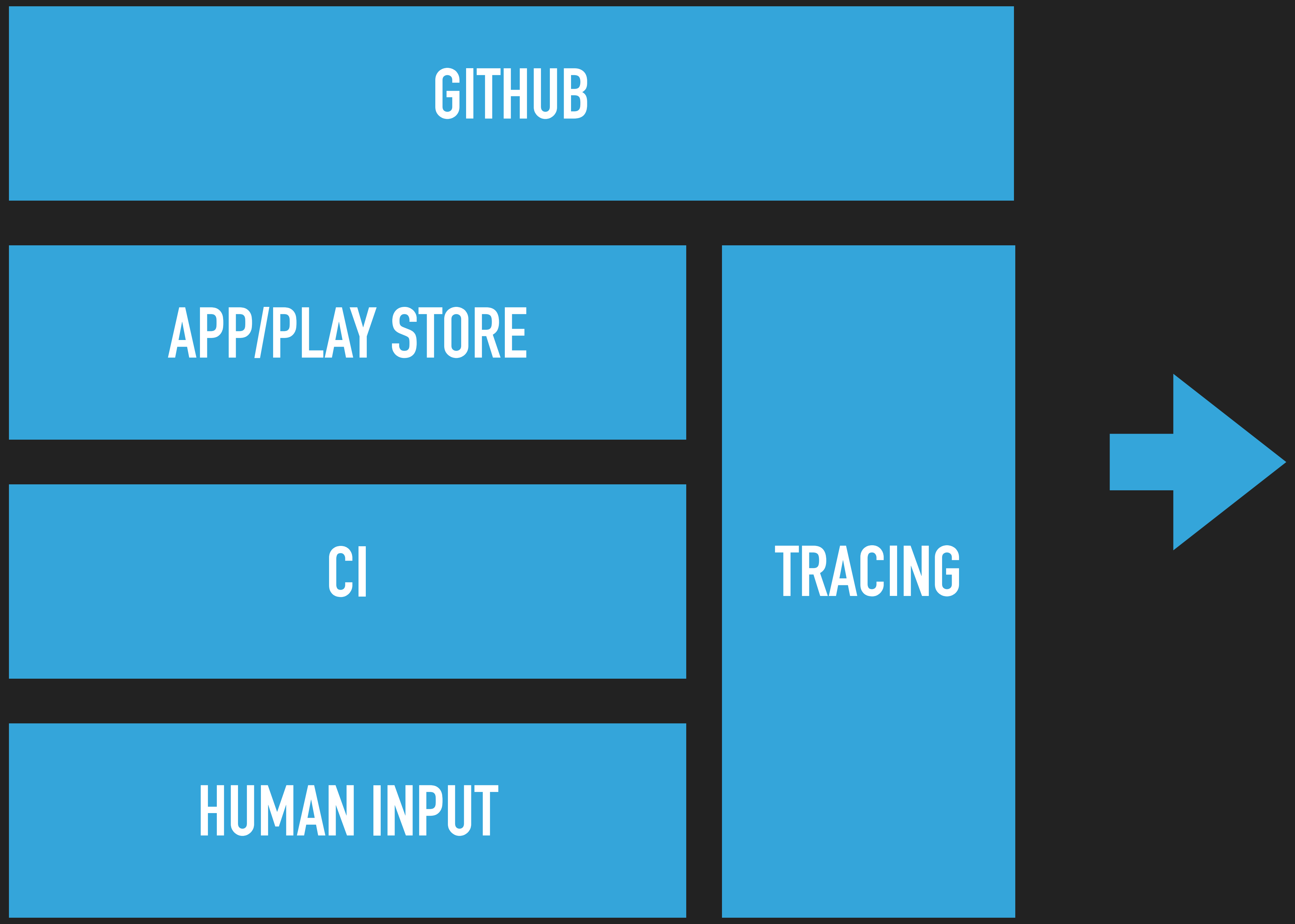
CELEBRATE

MITIGATE RISK

COMMUNICATE, COMMUNICATE, COMMUNICATE

THE PROCESS OF CHANGE

INSTRUMENT THE PROCESS

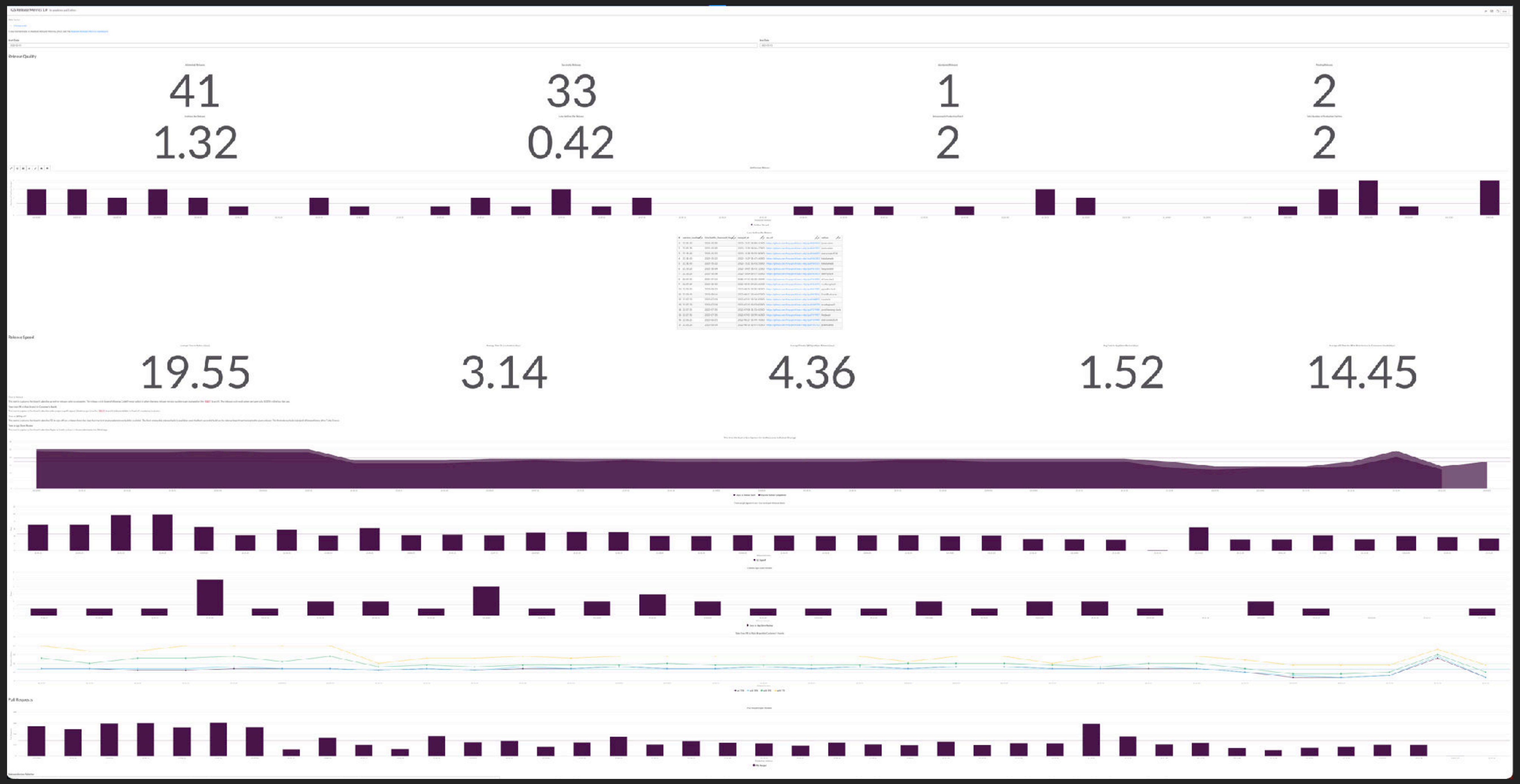

GITHUB

APP/PLAY STORE

CI

HUMAN INPUT

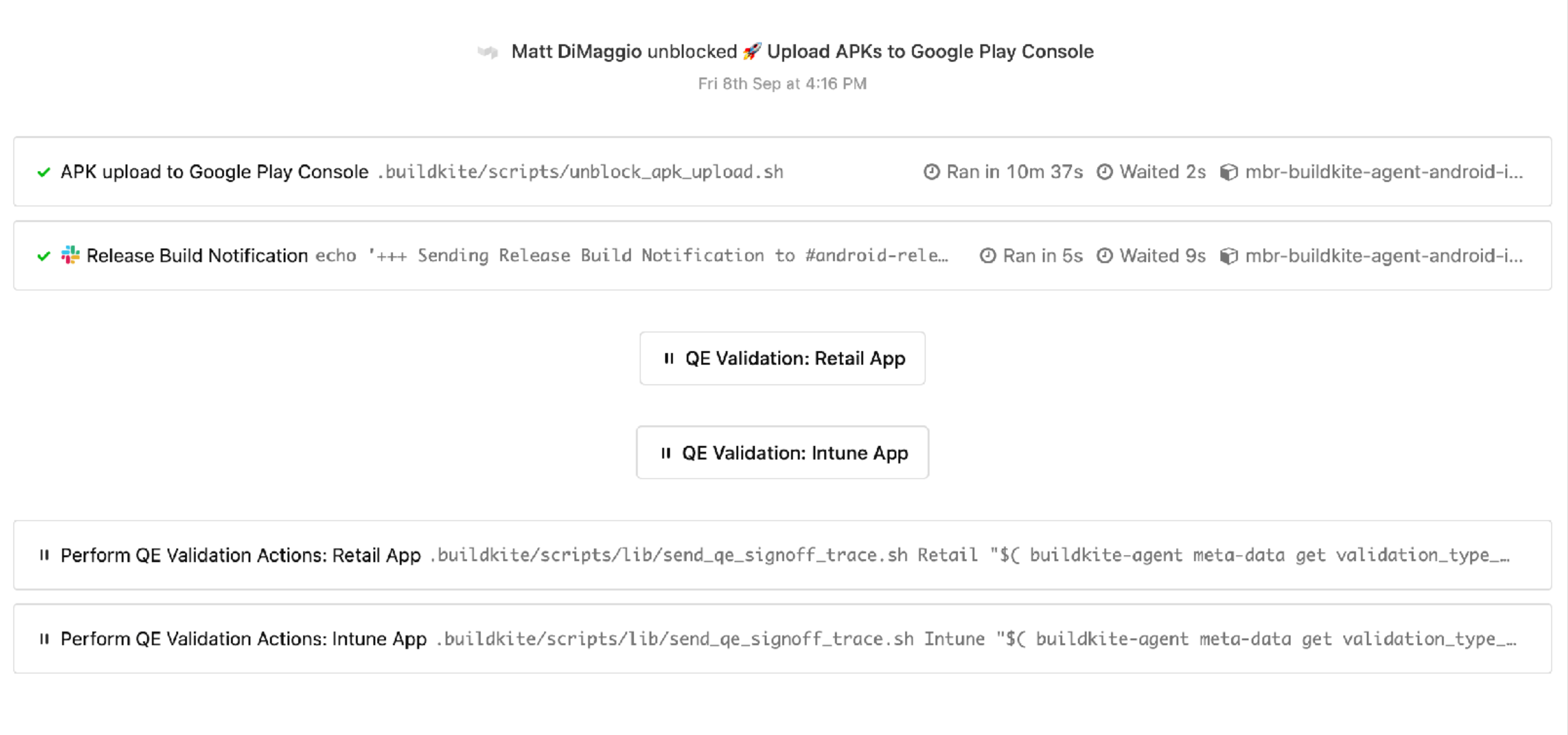
TRACING



THE PROCESS OF CHANGE

INSTRUMENT THE PROCESS

HUMAN INPUT



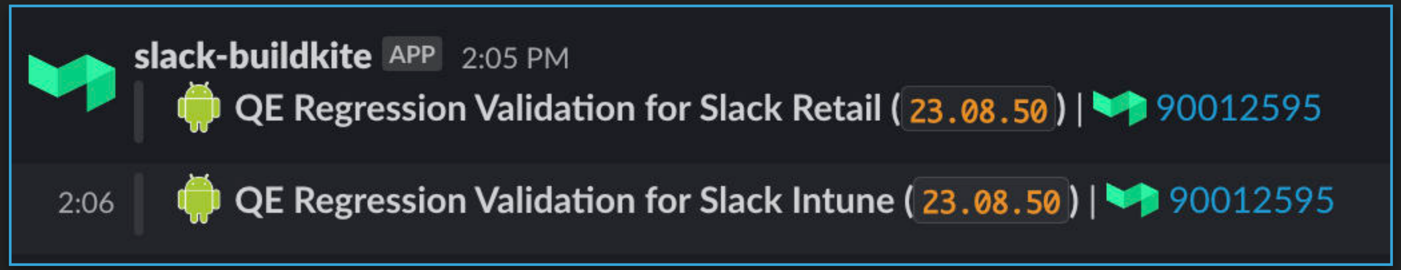
Matt DiMaggio unblocked 🚀 Upload APKs to Google Play Console
Fri 8th Sep at 4:16 PM

✓ APK upload to Google Play Console `.buildkite/scripts/unblock_apk_upload.sh` Ran in 10m 37s Waited 2s mbr-buildkite-agent-android-i...

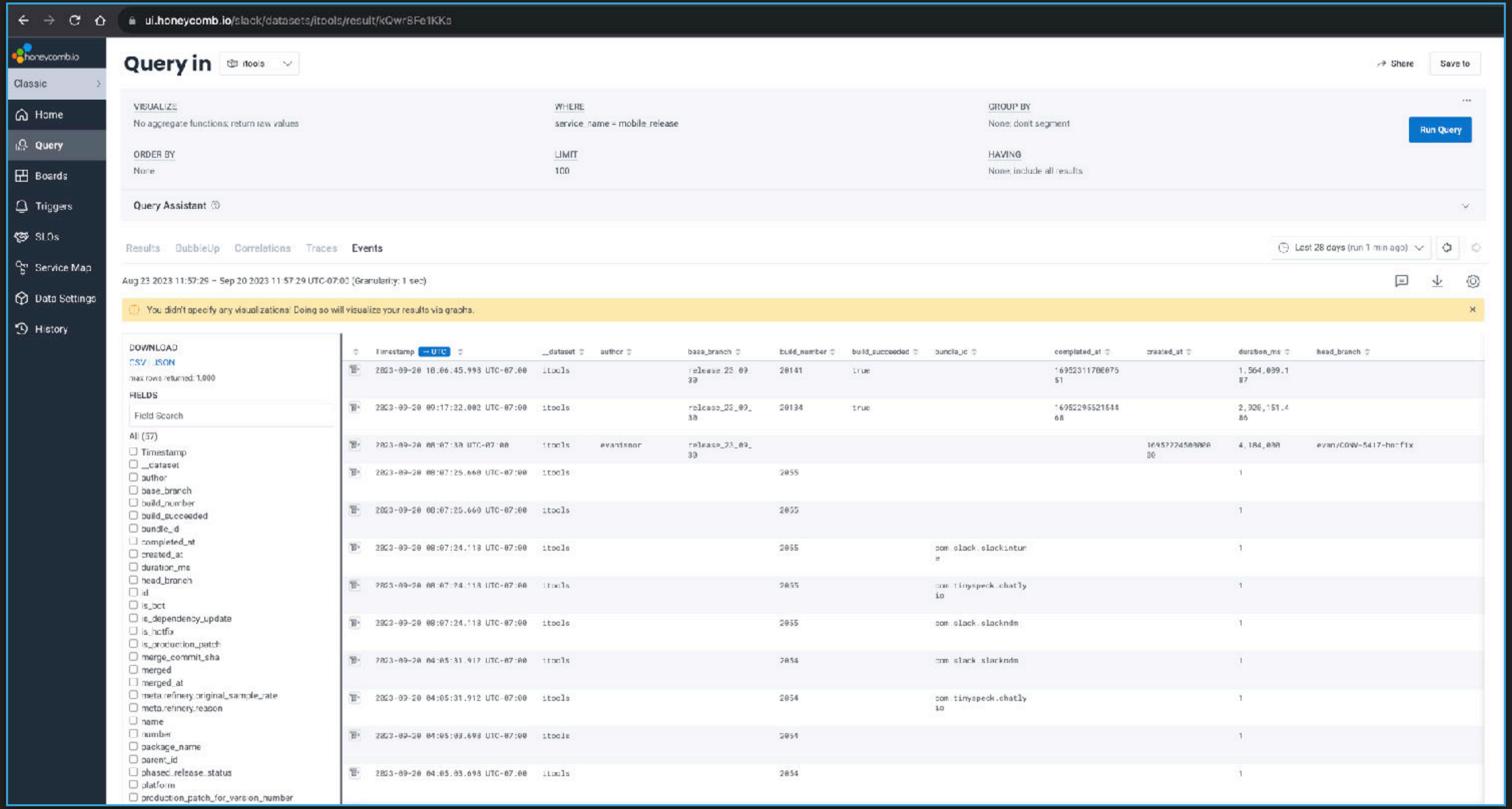
✓ Release Build Notification `echo '+++ Sending Release Build Notification to #android-rele...` Ran in 5s Waited 9s mbr-buildkite-agent-android-i...

⏸ Perform QE Validation Actions: Retail App `.buildkite/scripts/lib/send_qe_signoff_trace.sh Retail "$( buildkite-agent meta-data get validation_type_...`

⏸ Perform QE Validation Actions: Intune App `.buildkite/scripts/lib/send_qe_signoff_trace.sh Intune "$( buildkite-agent meta-data get validation_type_...`



slack-buildkite APP 2:05 PM
QE Regression Validation for Slack Retail (`23.08.50`) | 90012595
2:06 QE Regression Validation for Slack Intune (`23.08.50`) | 90012595



Query in itools

VISUALIZE
No aggregate functions; return raw values

WHERE
service_name = mobile_release

GROUP BY
None; don't segment

ORDER BY
None

LIMIT
100

HAVING
None; include all results

Query Assistant

Aug 23 2023 11:57:29 – Sep 20 2023 11:57:29 UTC-07:00 (Granularity: 1 sec)

You didn't specify any visualizations! Doing so will visualize your results via graphs.

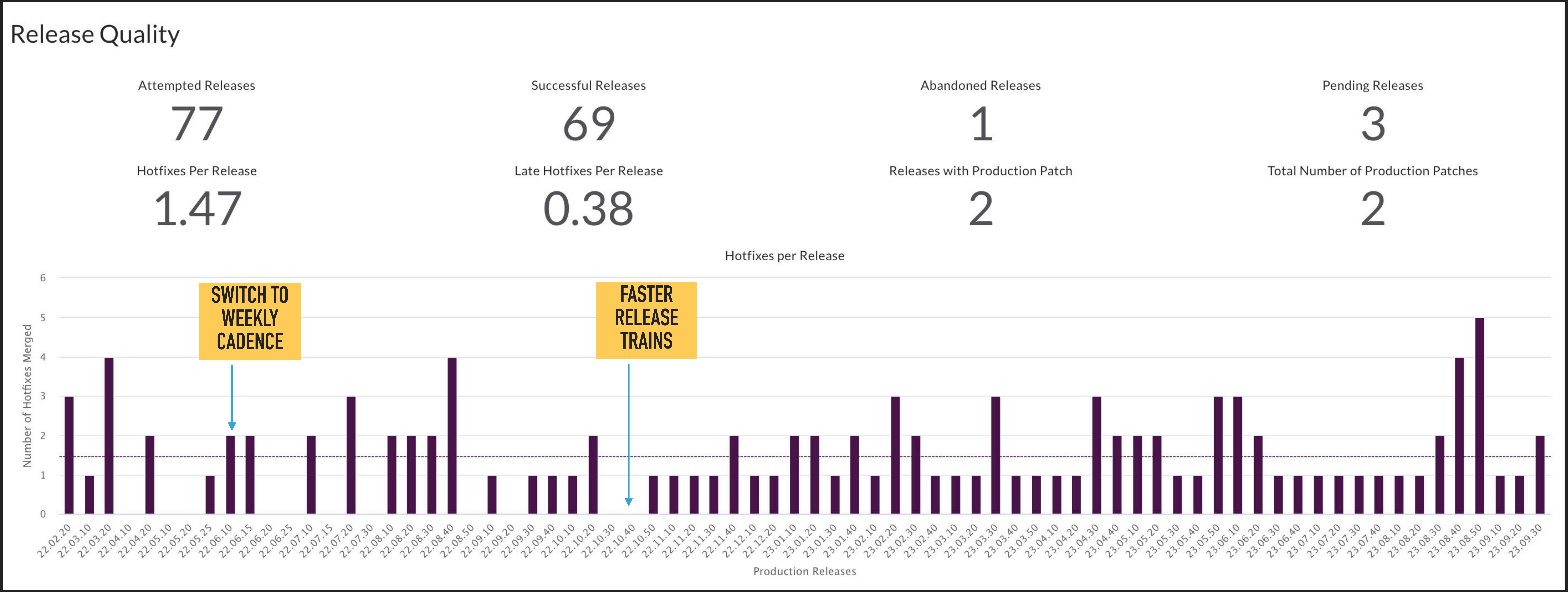
DOWNLOAD
CSV | JSON
max rows returned: 1,000

FIELDS
Field Search
All (37)
☐ Timestamp
☐ __dataset
☐ author
☐ base_branch
☐ build_number
☐ build_succeeded
☐ bundle_id
☐ completed_at
☐ created_at
☐ duration_ms
☐ head_branch
☐ id
☐ is_bot
☐ is_dependency_update
☐ is_hotfix
☐ is_production_patch
☐ merge_commit_sha
☐ merged
☐ merged_at
☐ meta.refinery.original_sample_rate
☐ meta.refinery.reason
☐ name
☐ number
☐ package_name
☐ parent_id
☐ phased_release_status
☐ platform
☐ production_patch_for_version_number

Timestamp → UTC	__dataset	author	base_branch	build_number	build_succeeded	bundle_id	completed_at	created_at	duration_ms	head_branch
2023-09-20 10:06:45.993 UTC-07:00	itools		release_23_09_30	20141	true		1695231178067551		1,564,039.187	
2023-09-20 09:17:22.002 UTC-07:00	itools		release_23_09_30	20134	true		1695229552154468		2,926,151.486	
2023-09-20 08:07:30 UTC-07:00	itools	evanisnor	release_23_09_30					1695222450000000	4,184,000	evan/CONV-5417-hotfix
2023-09-20 08:07:25.660 UTC-07:00	itools			2055					1	
2023-09-20 08:07:25.660 UTC-07:00	itools			2055					1	
2023-09-20 08:07:24.118 UTC-07:00	itools			2055		com.slack.slackintune			1	
2023-09-20 08:07:24.118 UTC-07:00	itools			2055		com.tinyspeck.chatlyio			1	
2023-09-20 08:07:24.118 UTC-07:00	itools			2055		com.slack.slackndm			1	
2023-09-20 04:05:31.912 UTC-07:00	itools			2054		com.slack.slackndm			1	
2023-09-20 04:05:31.912 UTC-07:00	itools			2054		com.tinyspeck.chatlyio			1	
2023-09-20 04:05:03.698 UTC-07:00	itools			2054					1	
2023-09-20 04:05:03.698 UTC-07:00	itools			2054					1	

THE PROCESS OF CHANGE

INSTRUMENT THE PROCESS



Release Quality

Attempted Releases	Successful Releases	Abandoned Releases	Pending Releases
77	69	1	3
Hotfixes Per Release	**Late Hotfixes Per Release**	**Releases with Production Patch**	**Total Number of Production Patches**
1.47	0.38	2	2

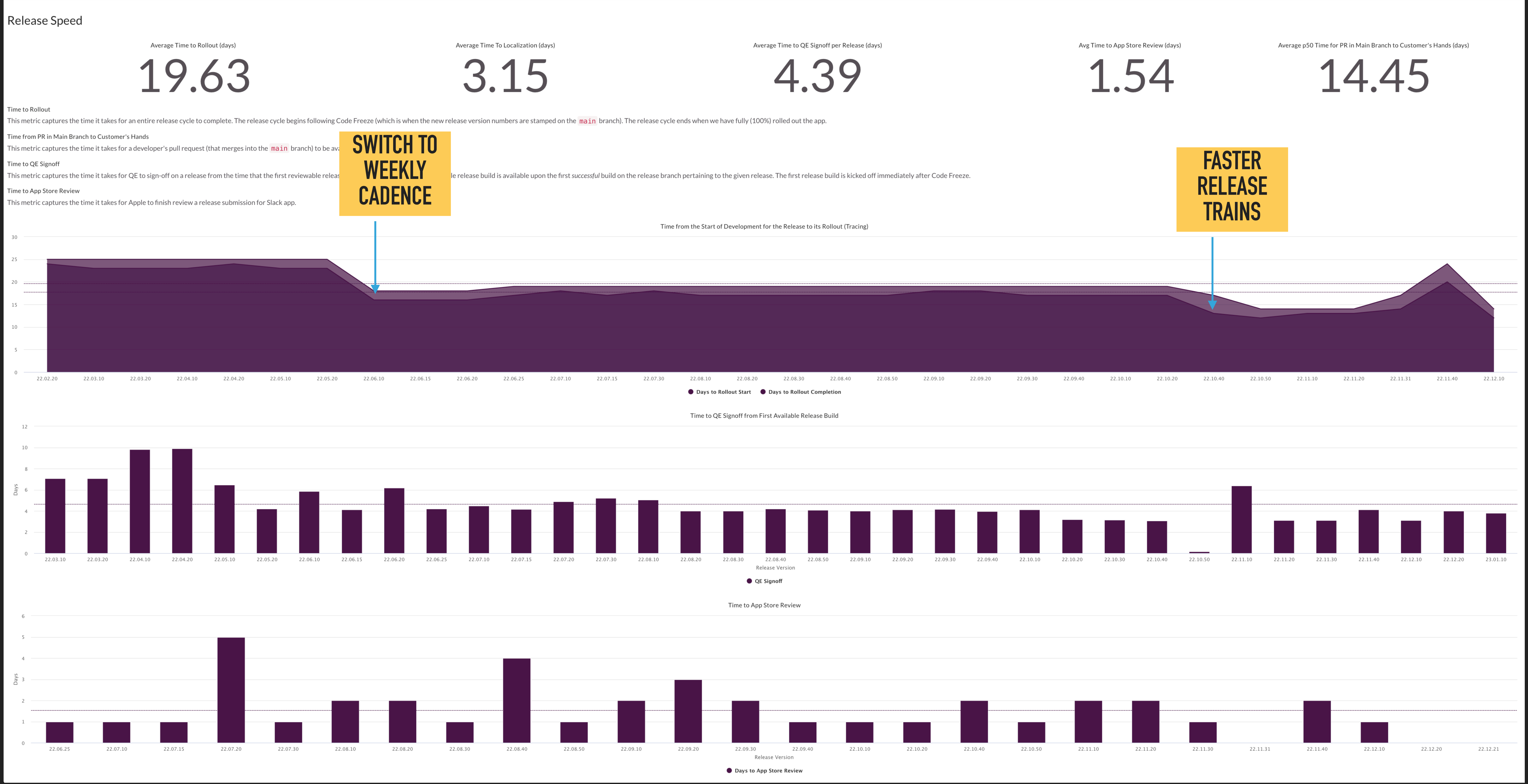
Hotfixes per Release

SWITCH TO WEEKLY CADENCE

FASTER RELEASE TRAINS

THE PROCESS OF CHANGE

INSTRUMENT THE PROCESS



Release Speed

Average Time to Rollout (days)	Average Time To Localization (days)	Average Time to QE Signoff per Release (days)	Avg Time to App Store Review (days)	Average p50 Time for PR in Main Branch to Customer's Hands (days)
19.63	3.15	4.39	1.54	14.45

Time to Rollout

This metric captures the time it takes for an entire release cycle to complete. The release cycle begins following Code Freeze (which is when the new release version numbers are stamped on the `main` branch). The release cycle ends when we have fully (100%) rolled out the app.

Time from PR in Main Branch to Customer's Hands

This metric captures the time it takes for a developer's pull request (that merges into the `main` branch) to be ava

Time to QE Signoff

This metric captures the time it takes for QE to sign-off on a release from the time that the first reviewable releas le release build is available upon the first *successful* build on the release branch pertaining to the given release. The first release build is kicked off immediately after Code Freeze.

Time to App Store Review

This metric captures the time it takes for Apple to finish review a release submission for Slack app.

SWITCH TO WEEKLY CADENCE

FASTER RELEASE TRAINS

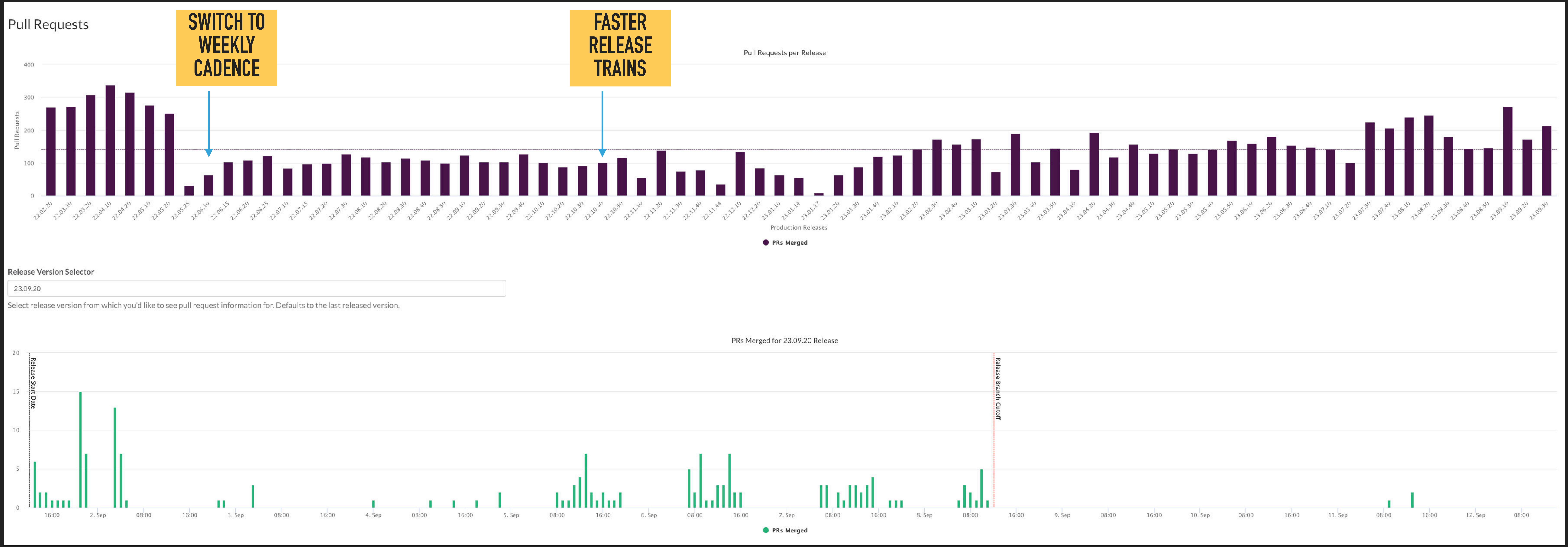
Time from the Start of Development for the Release to its Rollout (Tracing)

Time to QE Signoff from First Available Release Build

Time to App Store Review

THE PROCESS OF CHANGE

INSTRUMENT THE PROCESS



SWITCH TO WEEKLY CADENCE

FASTER RELEASE TRAINS

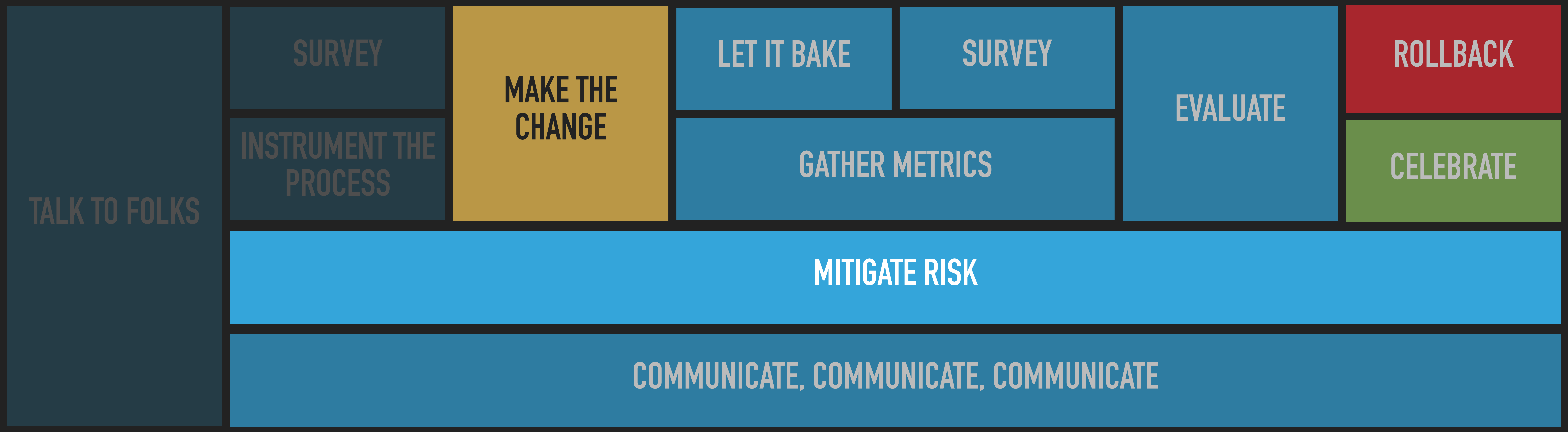
Pull Requests

Release Version Selector

23.09.20

Select release version from which you'd like to see pull request information for. Defaults to the last released version.

THE PROCESS OF CHANGE



TALK TO FOLKS

SURVEY

INSTRUMENT THE PROCESS

MAKE THE CHANGE

LET IT BAKE

SURVEY

GATHER METRICS

EVALUATE

ROLLBACK

CELEBRATE

MITIGATE RISK

COMMUNICATE, COMMUNICATE, COMMUNICATE

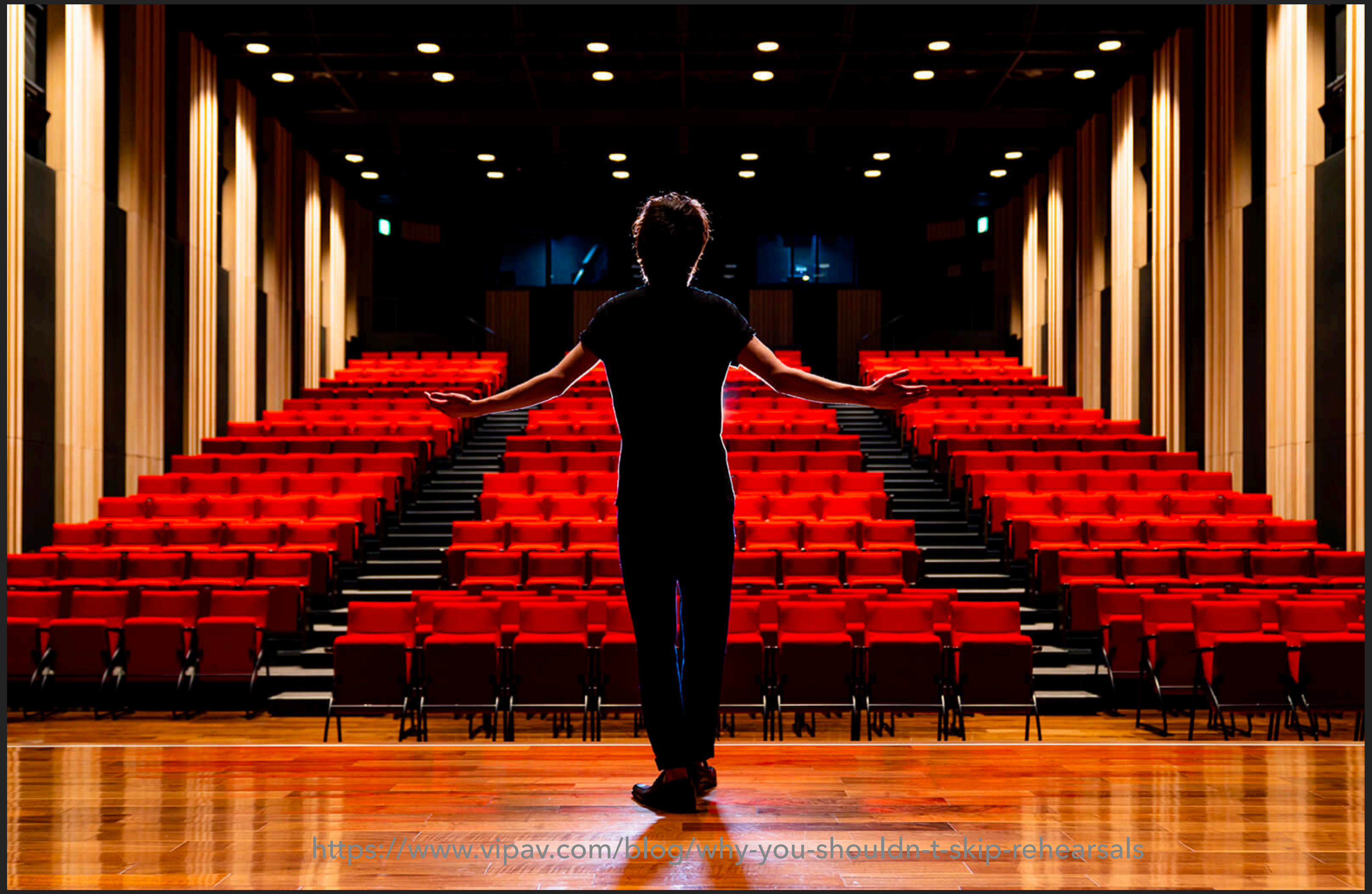
MITIGATE RISK

- Make the change incrementally
  - Stage 1: Increase release frequency
  - Stage 2: Increase release speed

https://www.pocketmindfulness.com/baby-steps-approach/

MITIGATE RISK

- Practice with a dry-run
  - Go through the process, but skip rollout



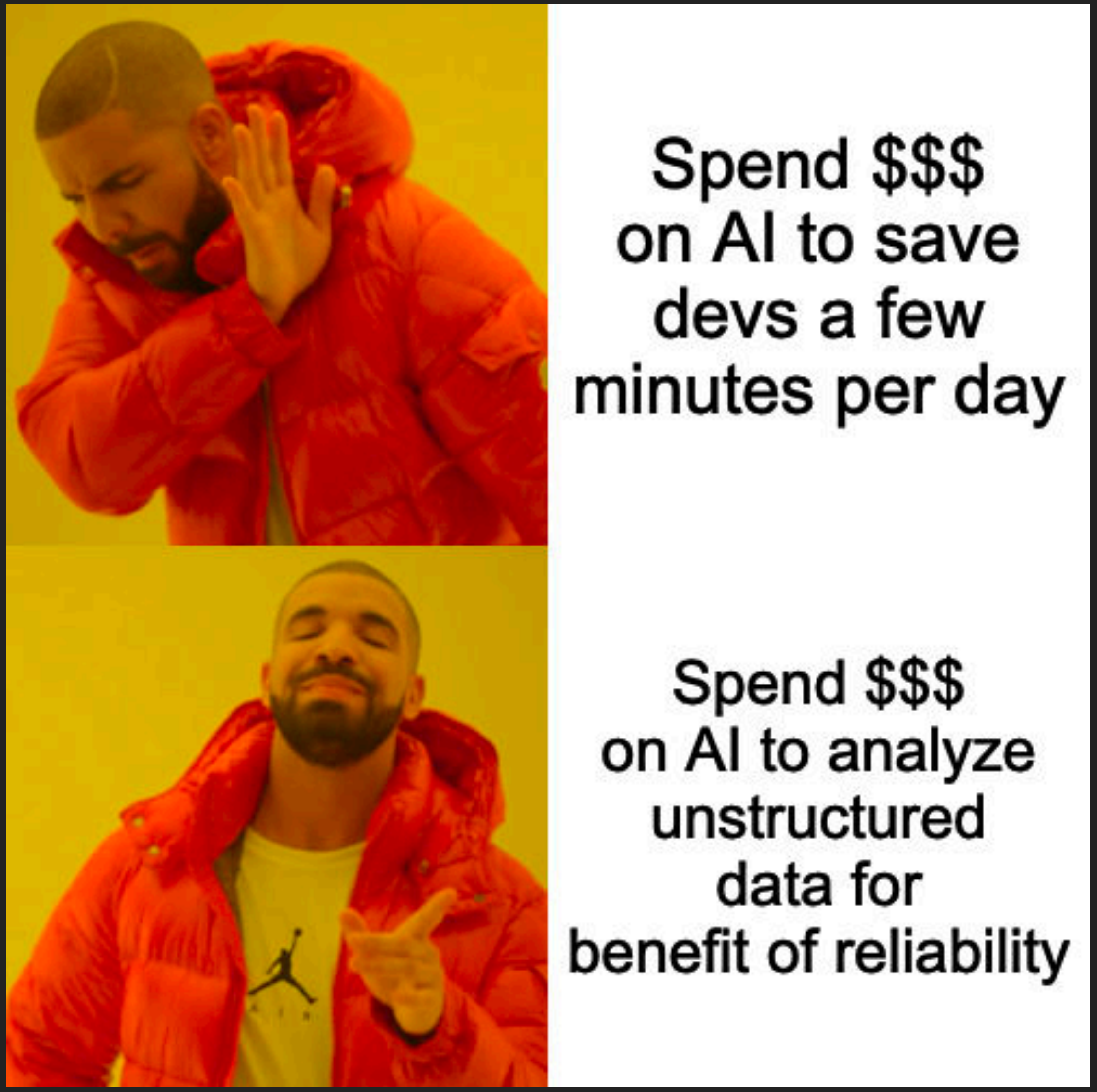
https://www.vipav.com/blog/why-you-shouldn-t-skip-rehearsals

THE PROCESS OF CHANGE

			Pressure	Schedule Clarity			Velocity			Safety	
			I/my team feel(s) pressure to "catch the release train" around string/code freeze.	I have clarity around when a change merged into the main branch will be available to dogfood users.	I have clarity around when a change merged into the main branch will be available to external beta users.	I have clarity around when a change merged into the main branch will become available to all external users.	I am satisfied with the speed at which changes from main branch become available to dogfood users.	I am satisfied with the speed at which changes from main branch become available to external beta users.	I am satisfied with the speed at which changes from main branch become available to all external users.	I am comfortable with releasing "green" builds from main branch to all external users without waiting for results of the manual regression test pass.	Feature flagging is an effective way to mitigate risk for changes in our app.
		Average	3.50	3.53	2.88	3.57	3.57	3.28	3.14	2.50	4.07
Scale	Average by discipline	**Development**	3.14	3.86	2.95	3.81	3.95	3.43	3.67	2.48	4.14
5 = Strongly Agree		**Design**	3.80	2.60	2.60	2.80	3.00	3.00	2.60	3.00	3.60
4 = Agree		**Engineering Management**	3.22	3.78	2.89	3.67	4.00	3.00	2.89	2.33	3.78
3 = Neutral		**Product Management**	4.00	2.79	2.36	3.36	2.79	2.93	2.64	2.79	4.36
2 = Disagree		**Quality Engineering**	3.71	4.00	3.29	3.29	3.71	3.86	3.29	2.14	4.14
1 = Strongly Disagree		**Program Management**	3.50	5.00	5.00	5.00	4.00	4.00	3.00	1.50	3.50
Legend	Average by Product/Infra	**Product Engineering**	3.81	3.60	2.95	3.67	3.55	3.36	3.21	2.55	4.24
🟥 bad		**Other**	3.33	2.67	2.50	2.50	3.00	3.00	2.33	2.67	3.33
🟩 green = good		**Infrastructure**	2.30	3.80	2.80	3.80	4.00	3.10	3.30	2.20	3.80
	Average by Platform	**Android**	3.25	3.63	3.25	3.94	4.00	3.69	3.69	2.38	4.25
		Android|iOS	3.89	3.33	2.81	3.56	3.22	3.07	2.74	2.52	4.07
		iOS	3.07	3.80	2.60	3.20	3.73	3.20	3.27	2.60	3.87

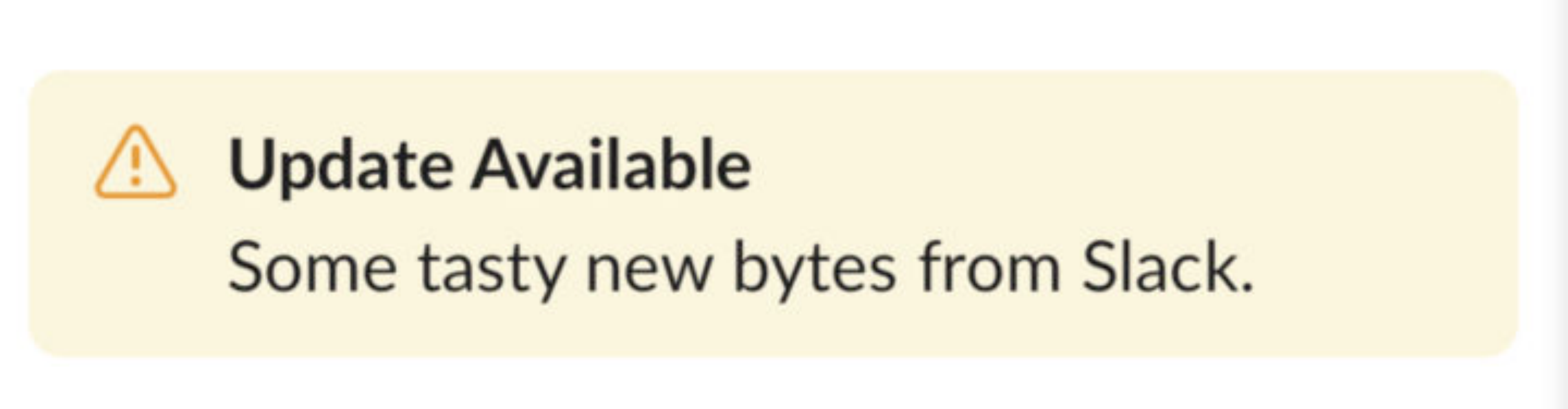
MITIGATE RISK

- Feature flags
  - Not a silver bullet
  - Measure % or change behind flag?

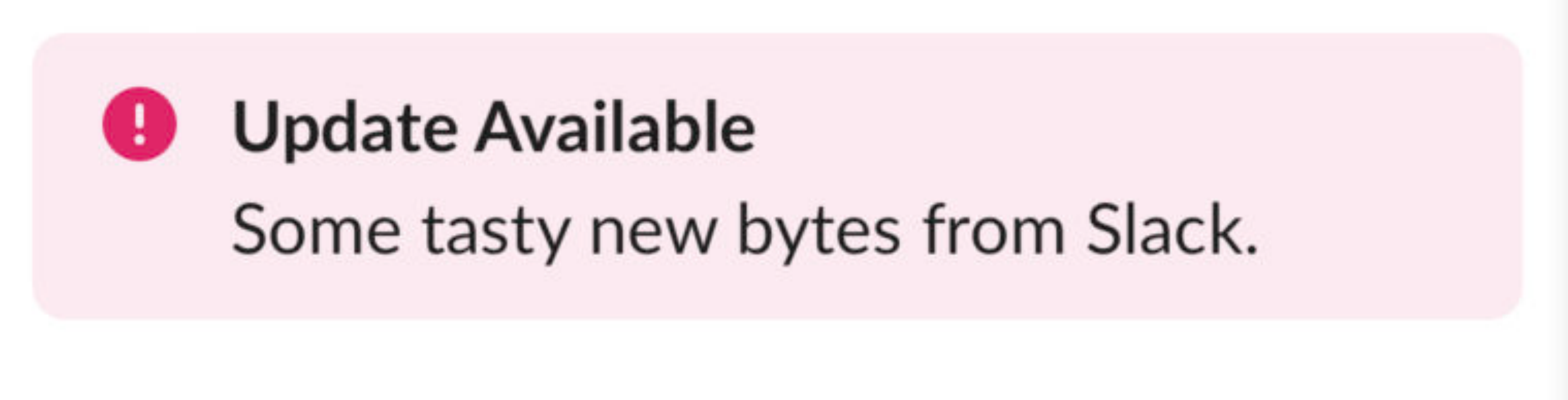


MITIGATE RISK

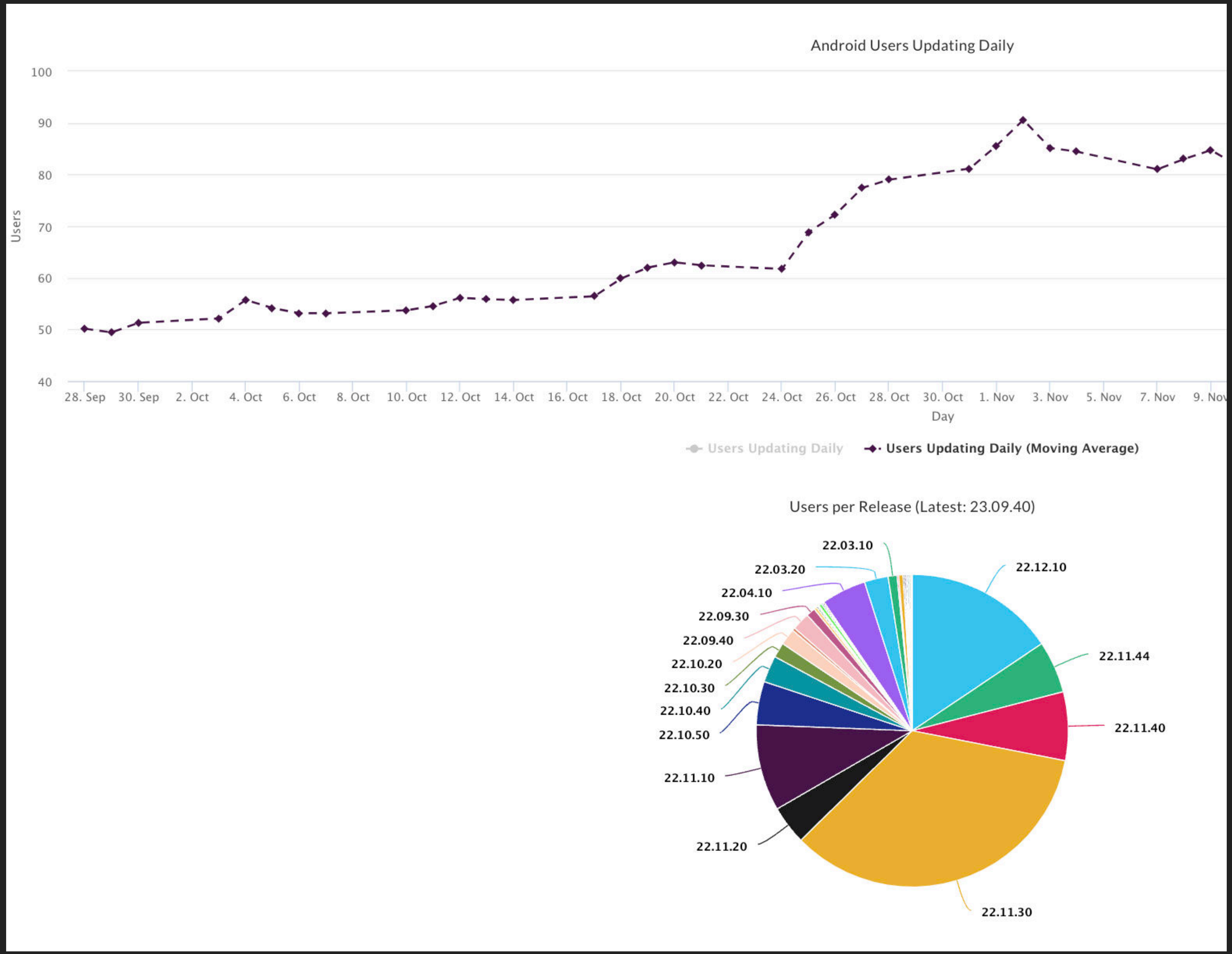
- Dogfood (internal testing)
  - Reduced time to catch errors
  - Compensated by increase in usage



⚠ **Update Available**
Some tasty new bytes from Slack.



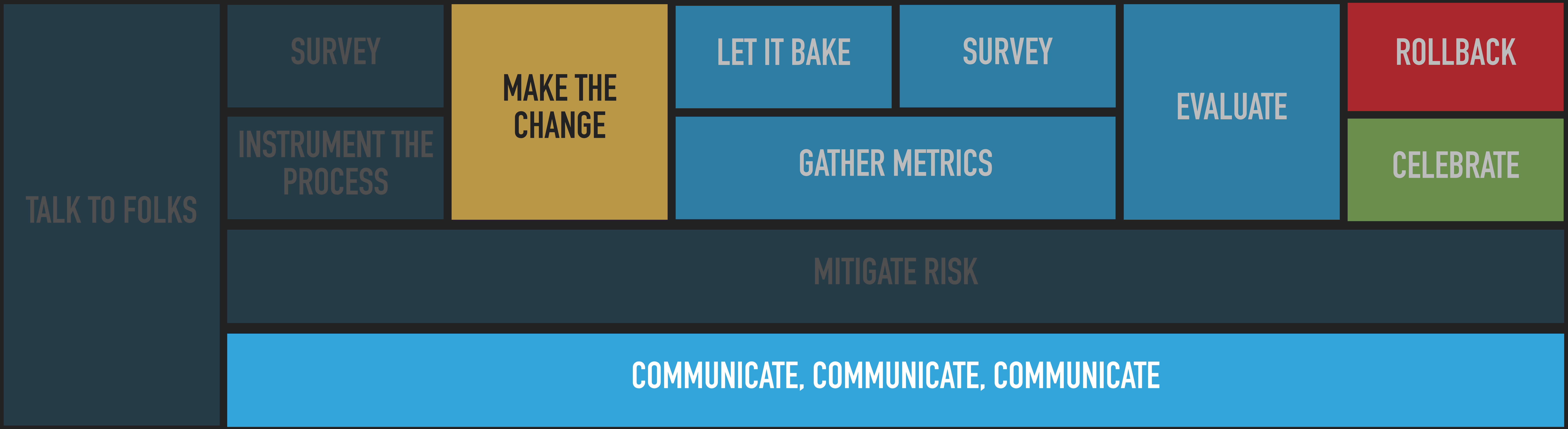
❗ **Update Available**
Some tasty new bytes from Slack.



THE PROCESS OF CHANGE

LIVE WITH RISK

THE PROCESS OF CHANGE



TALK TO FOLKS

SURVEY

INSTRUMENT THE PROCESS

MAKE THE CHANGE

LET IT BAKE

SURVEY

GATHER METRICS

EVALUATE

ROLLBACK

CELEBRATE

MITIGATE RISK

COMMUNICATE, COMMUNICATE, COMMUNICATE

COMMUNICATE, COMMUNICATE, COMMUNICATE

- Repetition
- Repetition
- Keep it Real
- Educate
- Emphasis: Experimentation and Success Criteria

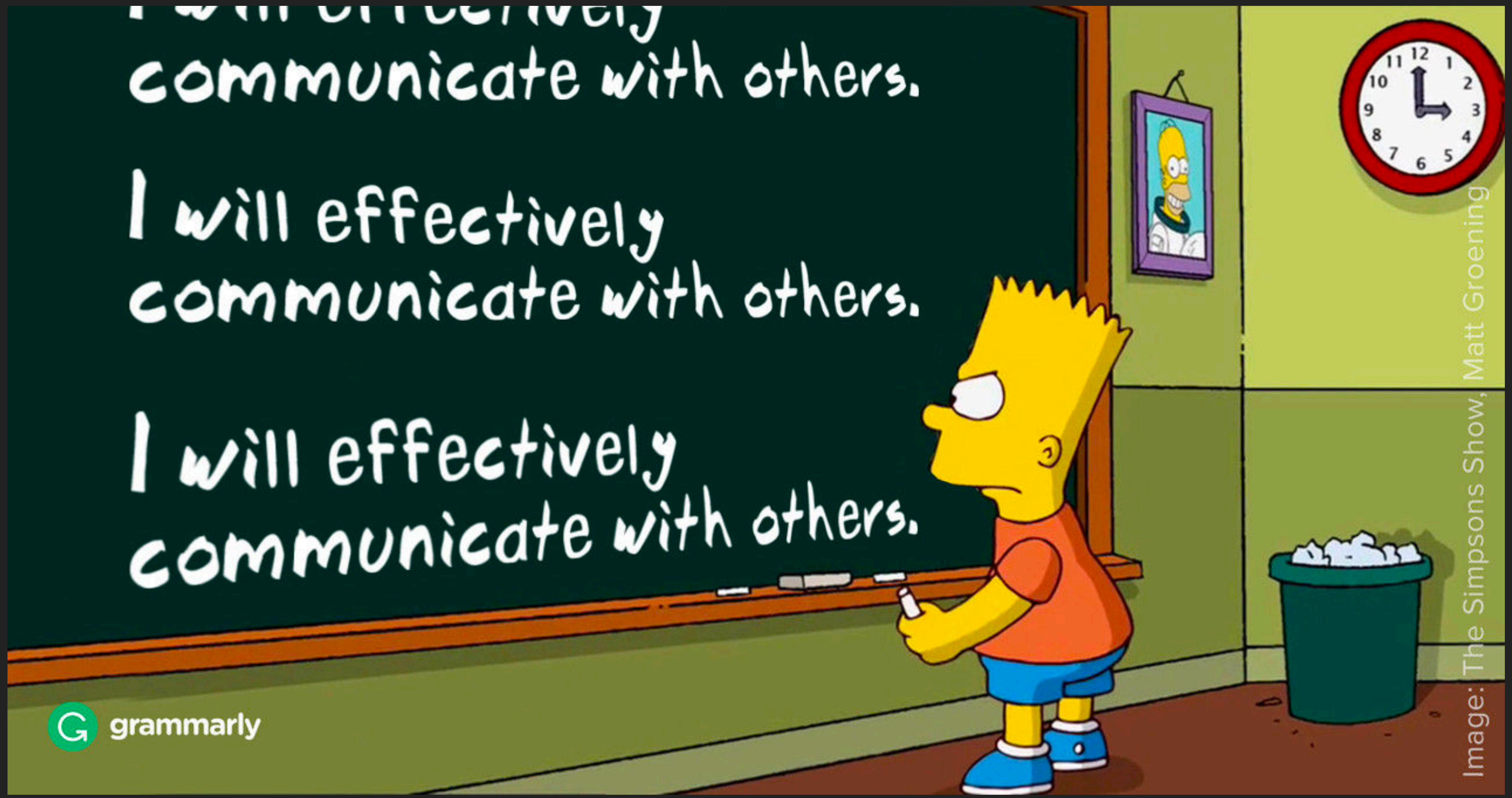
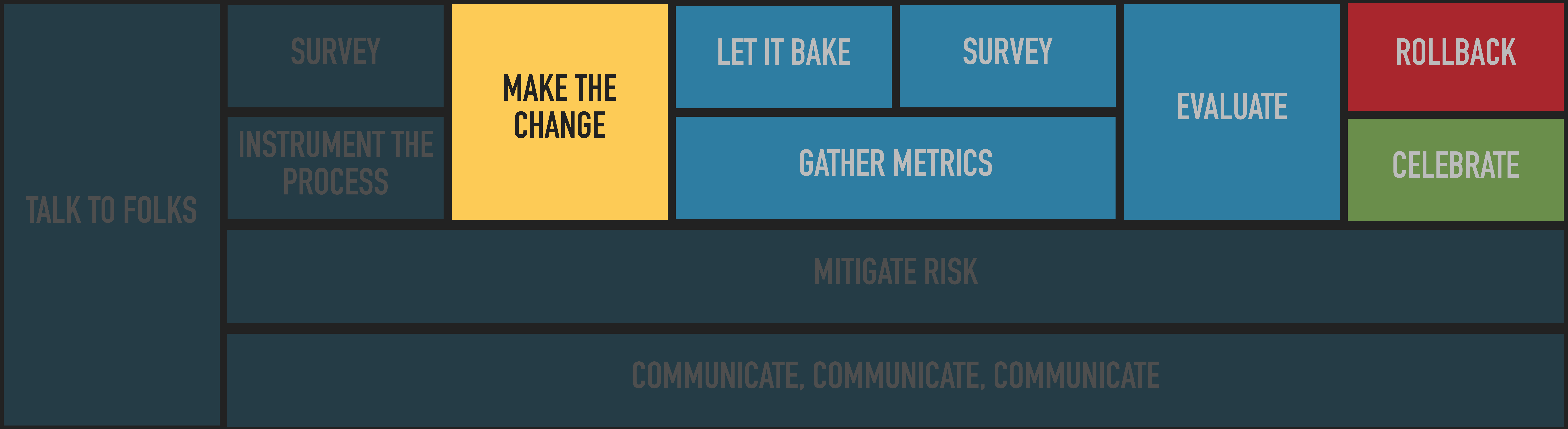


Image: The Simpsons Show, Matt Groening

THE PROCESS OF CHANGE



TALK TO FOLKS

SURVEY

INSTRUMENT THE PROCESS

MAKE THE CHANGE

LET IT BAKE

SURVEY

GATHER METRICS

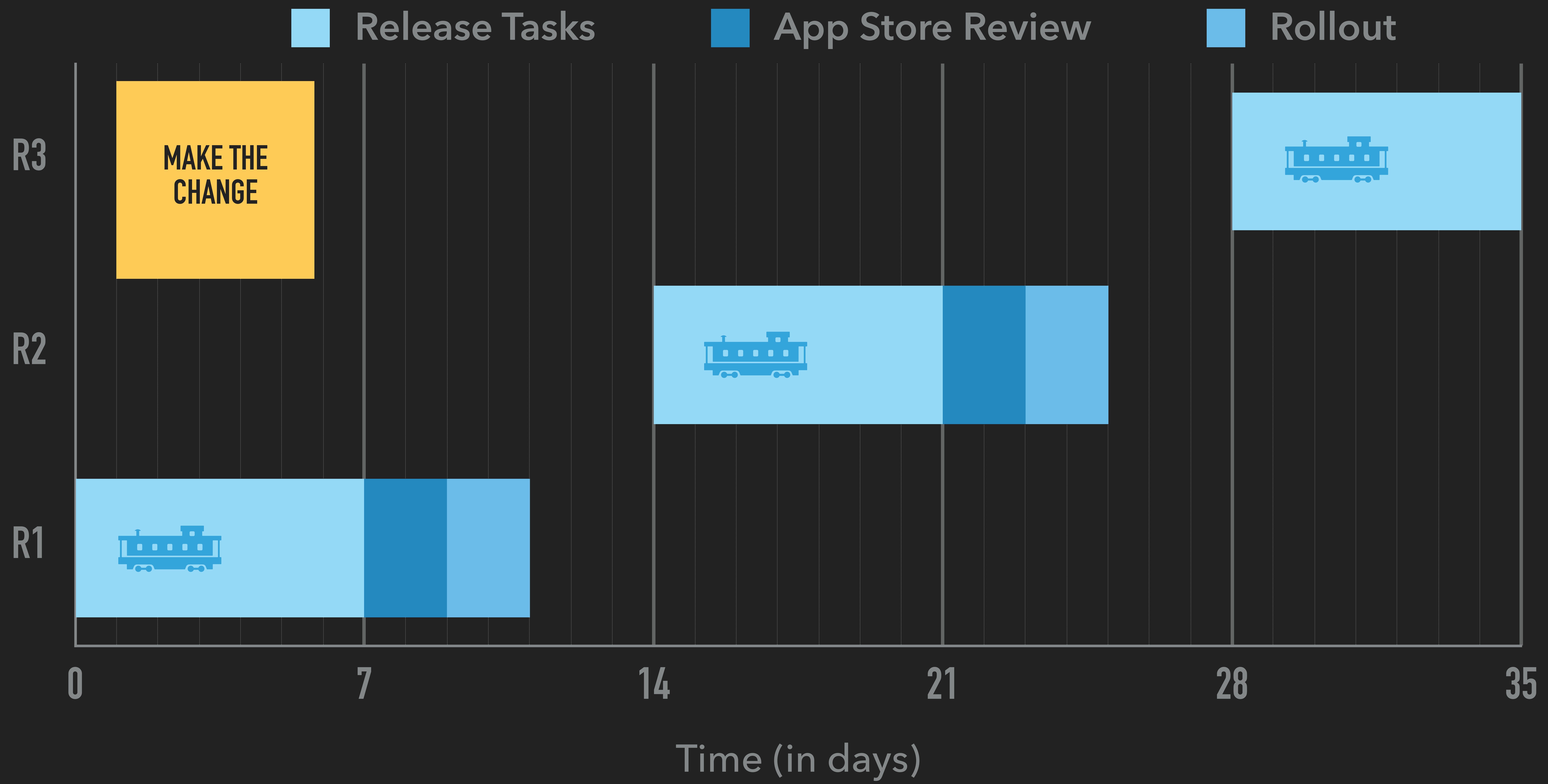
EVALUATE

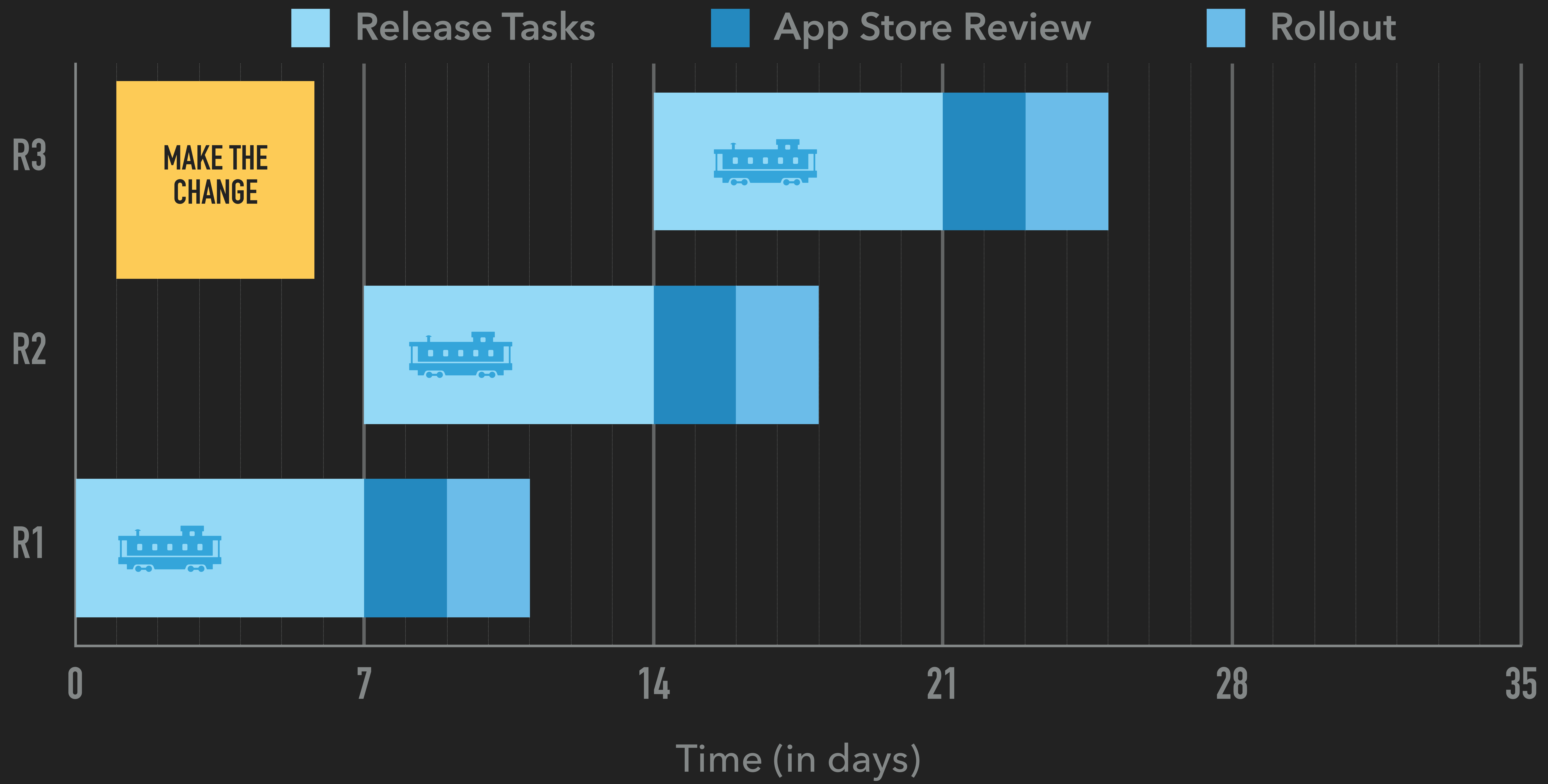
ROLLBACK

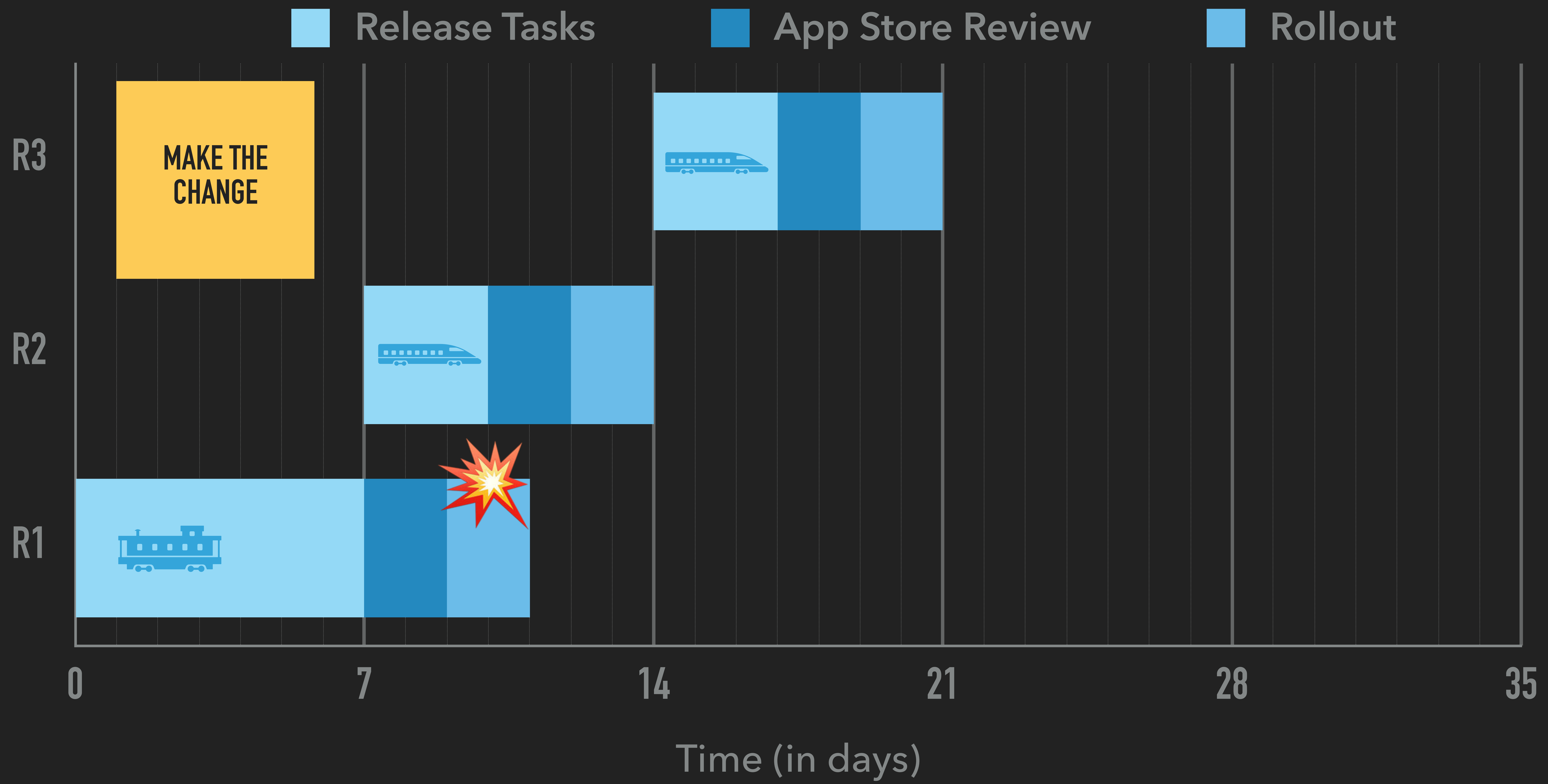
CELEBRATE

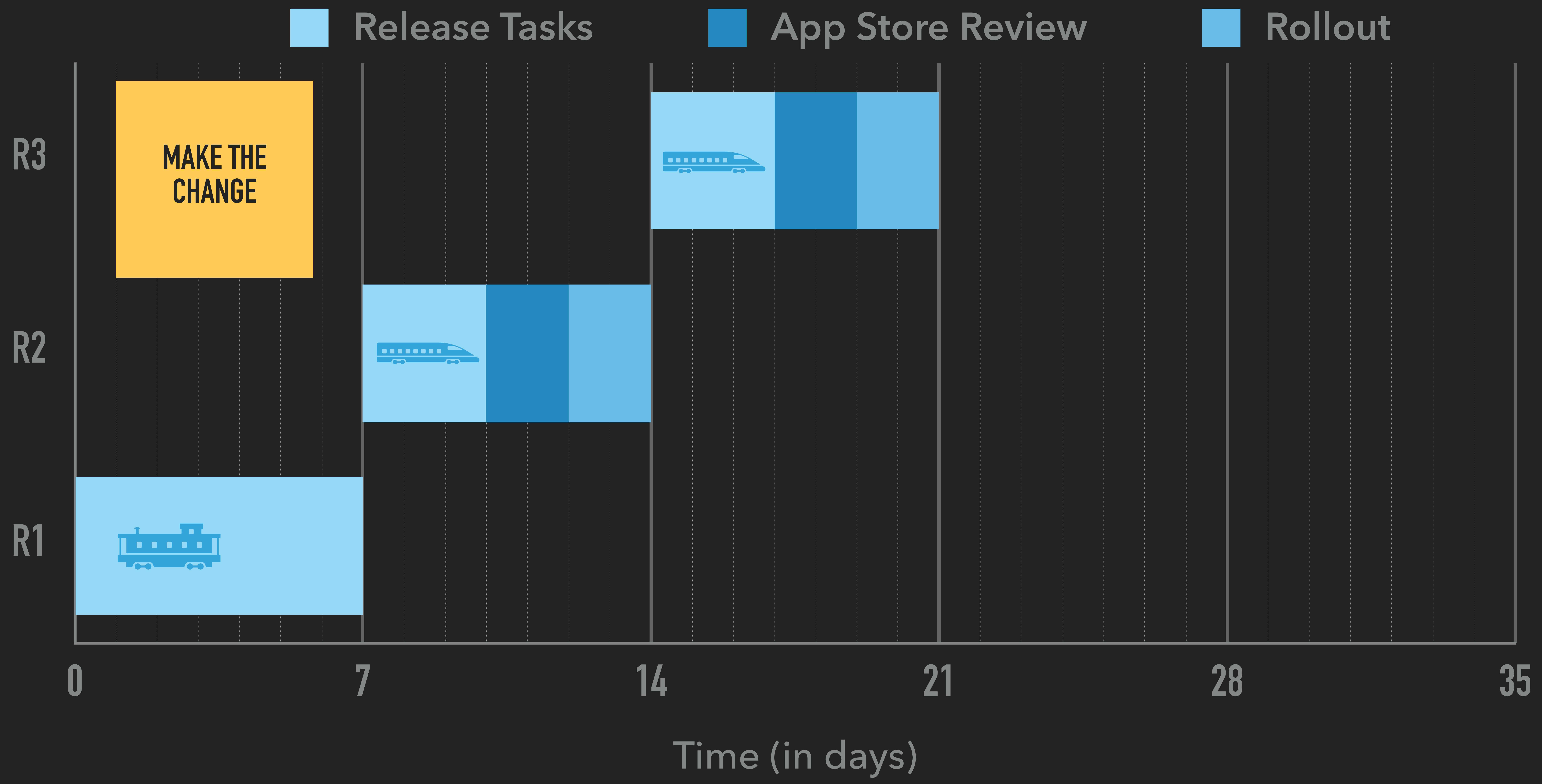
MITIGATE RISK

COMMUNICATE, COMMUNICATE, COMMUNICATE

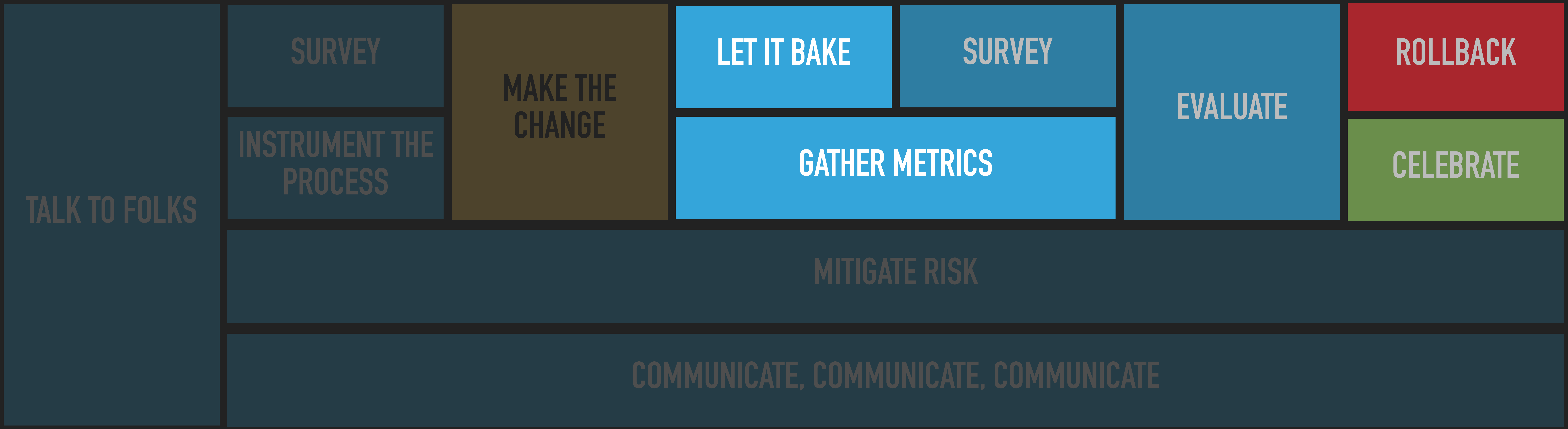

Release Tasks
App Store Review
Rollout
R3
MAKE THE CHANGE
R2
R1
0
7
14
21
28
35
Time (in days)


Release Tasks
App Store Review
Rollout
R3
MAKE THE CHANGE
R2
R1
0
7
14
21
28
35
Time (in days)


Release Tasks
App Store Review
Rollout
R3
R2
R1
MAKE THE CHANGE
0
7
14
21
28
35
Time (in days)


Release Tasks
App Store Review
Rollout
R3
R2
R1
MAKE THE CHANGE
0
7
14
21
28
35
Time (in days)

THE PROCESS OF CHANGE



TALK TO FOLKS

SURVEY

INSTRUMENT THE PROCESS

MAKE THE CHANGE

LET IT BAKE

SURVEY

GATHER METRICS

EVALUATE

ROLLBACK

CELEBRATE

MITIGATE RISK

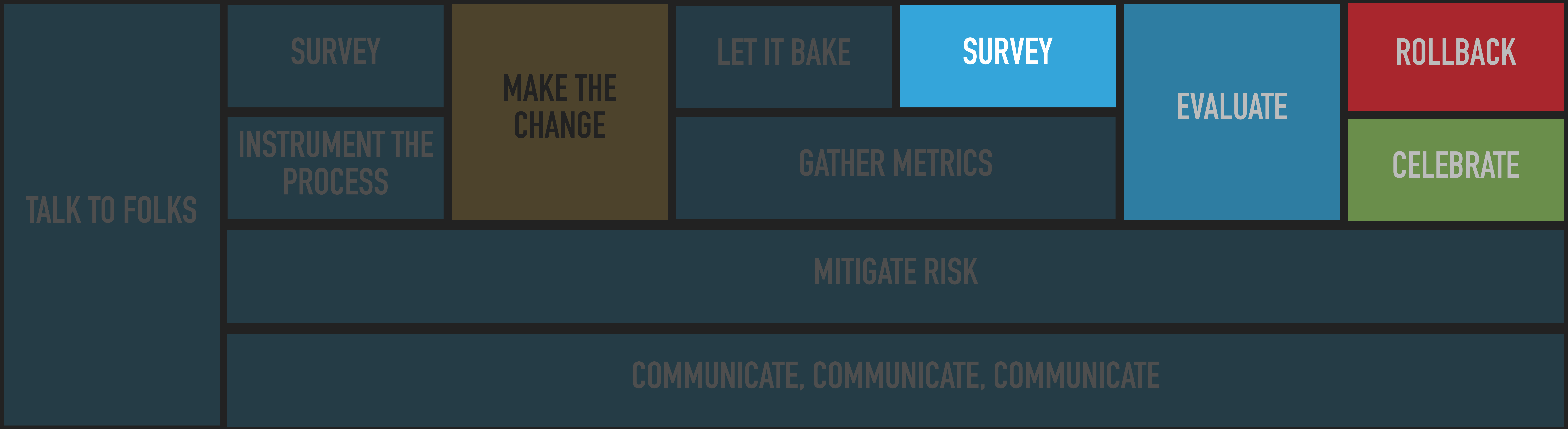
COMMUNICATE, COMMUNICATE, COMMUNICATE

THE PROCESS OF CHANGE

LET IT BAKE

GATHER METRICS

THE PROCESS OF CHANGE



TALK TO FOLKS

SURVEY

INSTRUMENT THE PROCESS

MAKE THE CHANGE

LET IT BAKE

SURVEY

GATHER METRICS

EVALUATE

ROLLBACK

CELEBRATE

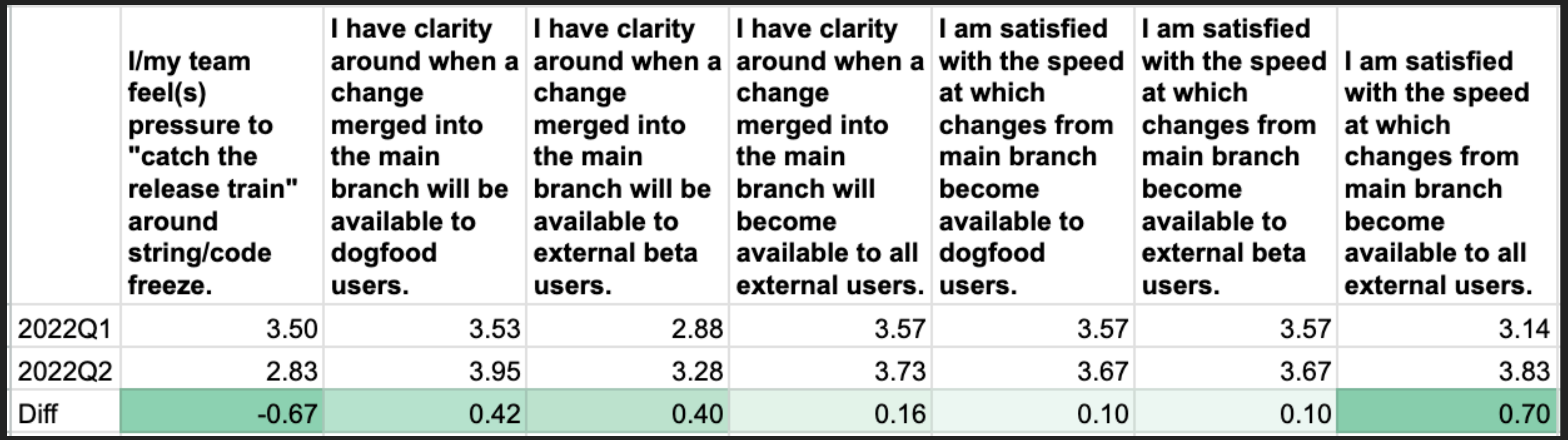
MITIGATE RISK

COMMUNICATE, COMMUNICATE, COMMUNICATE

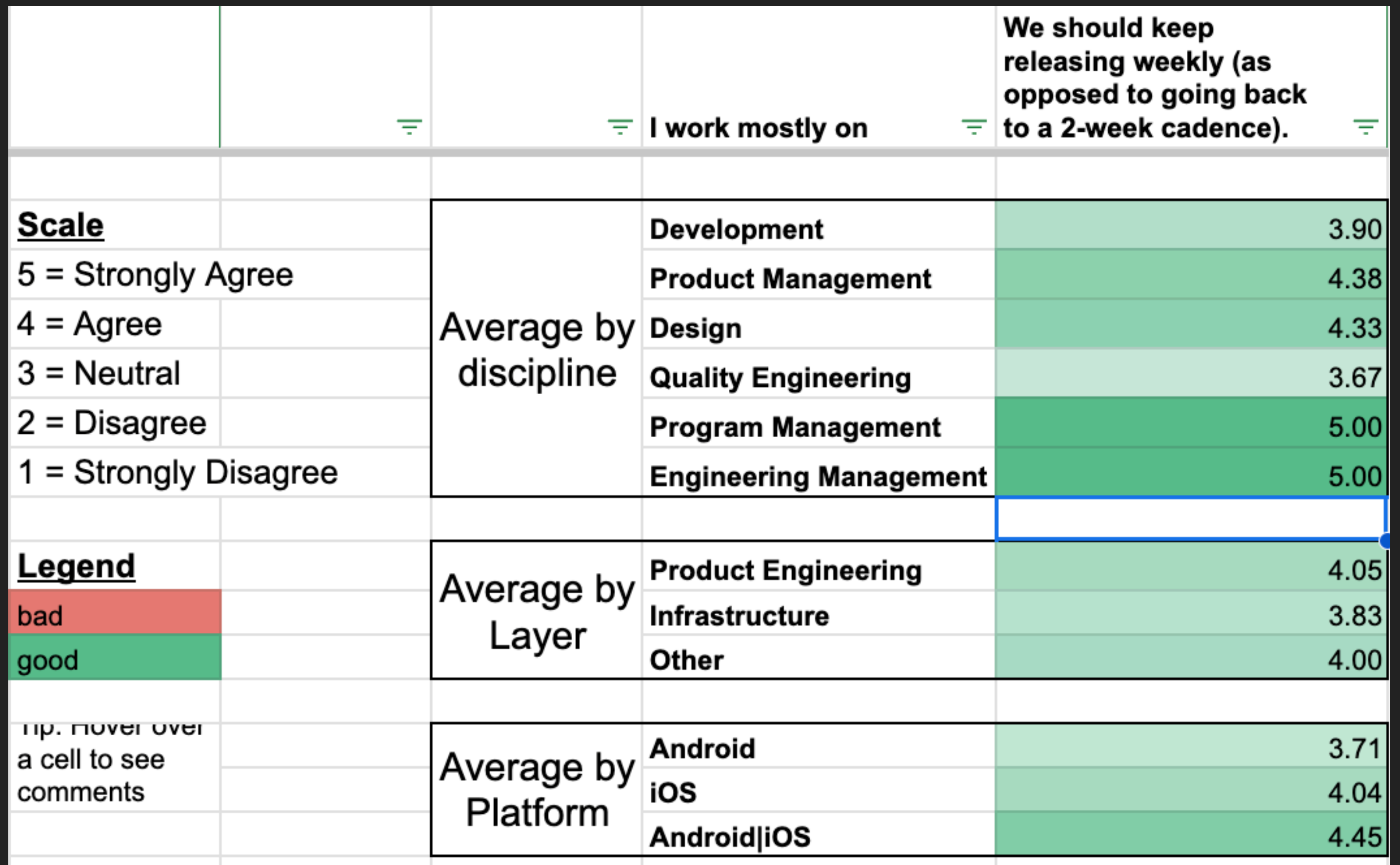
THE PROCESS OF CHANGE

AFTER SWITCH TO WEEKLY RELEASES

SURVEY



	I/my team feel(s) pressure to "catch the release train" around string/code freeze.	I have clarity around when a change merged into the main branch will be available to dogfood users.	I have clarity around when a change merged into the main branch will be available to external beta users.	I have clarity around when a change merged into the main branch will become available to all external users.	I am satisfied with the speed at which changes from main branch become available to dogfood users.	I am satisfied with the speed at which changes from main branch become available to external beta users.	I am satisfied with the speed at which changes from main branch become available to all external users.
2022Q1	3.50	3.53	2.88	3.57	3.57	3.57	3.14
2022Q2	2.83	3.95	3.28	3.73	3.67	3.67	3.83
Diff	-0.67	0.42	0.40	0.16	0.10	0.10	0.70



			I work mostly on	We should keep releasing weekly (as opposed to going back to a 2-week cadence).
Scale		Average by discipline	Development	3.90
5 = Strongly Agree			Product Management	4.38
4 = Agree			Design	4.33
3 = Neutral			Quality Engineering	3.67
2 = Disagree			Program Management	5.00
1 = Strongly Disagree			Engineering Management	5.00
Legend		Average by Layer	Product Engineering	4.05
🟥 bad			Infrastructure	3.83
🟩 good			Other	4.00
Tip: Hover over a cell to see comments		Average by Platform	Android	3.71
			iOS	4.04
			Android|iOS	4.45

> THIS USED TO BE A STRESSFUL REALITY, BUT THANKS TO OUR WEEKLY RELEASE SCHEDULE (AND THE MOVING OF STRING

A DEVELOPER

> WITHOUT PROJECT GROUNDHOG AND WEEKLY RELEASES, I FELT THE RELEASES WERE TOO INFREQUENT AND WE WOULD WAIT OR RESORT TO HOT FIXES MORE OFTEN.

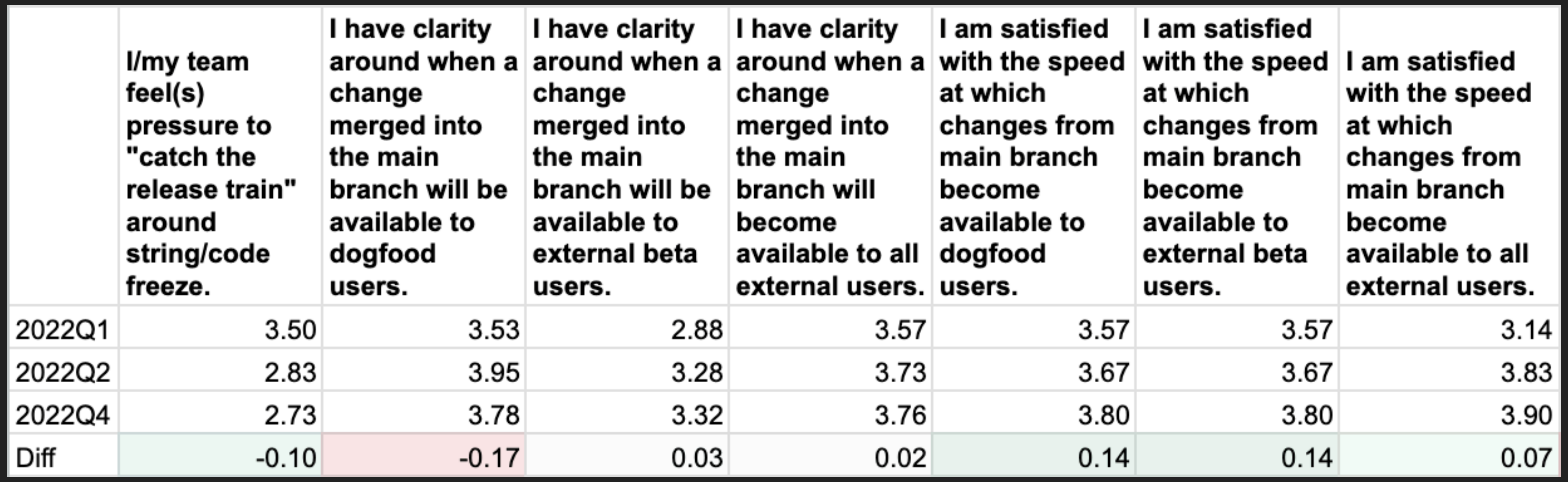
A DEVELOPER

> A WEEK AND A HALF FEELS LIKE A LONG TIME.

A DESIGNER

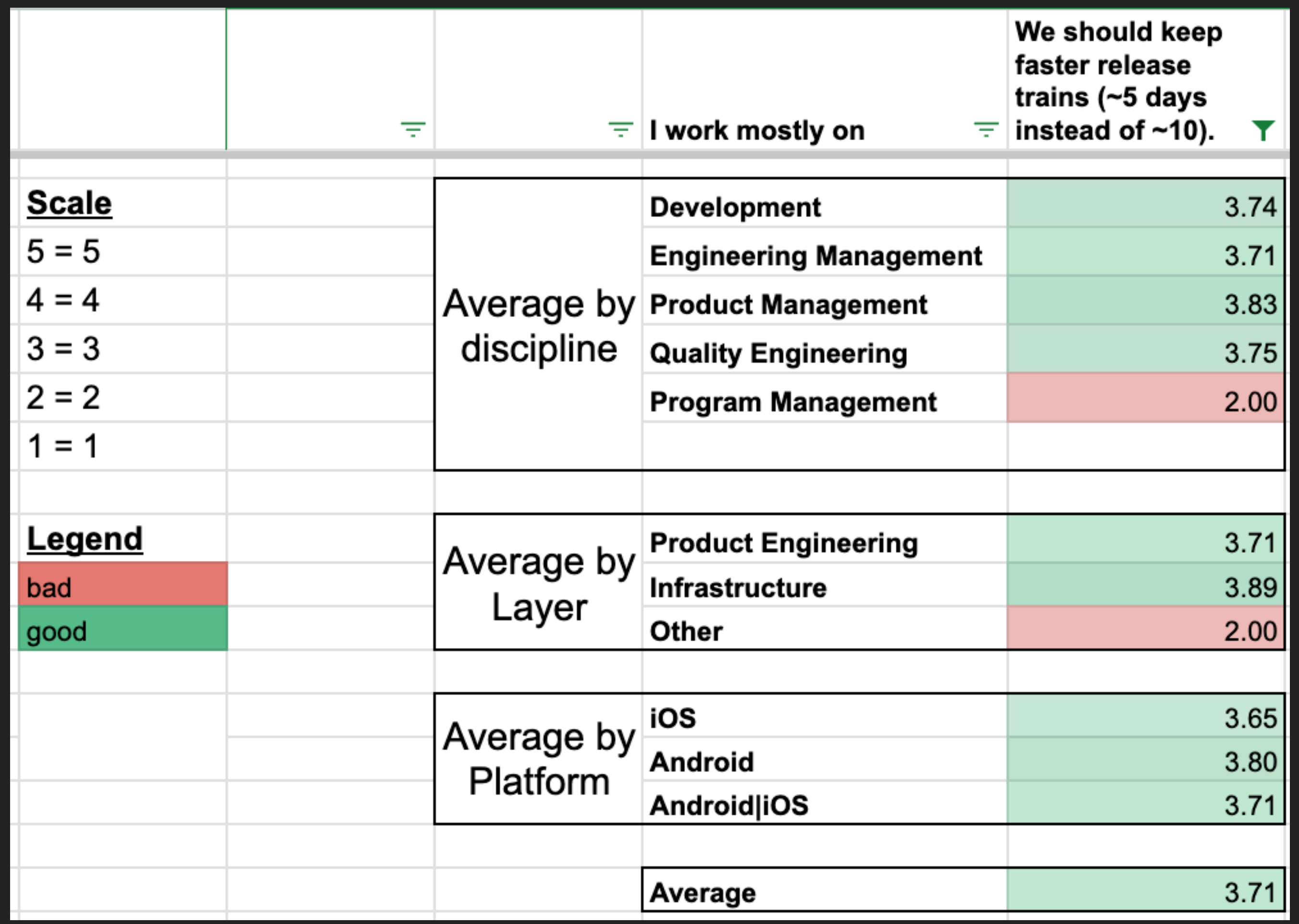
THE PROCESS OF CHANGE

AFTER SWITCH TO FASTER TRAIN



	I/my team feel(s) pressure to "catch the release train" around string/code freeze.	I have clarity around when a change merged into the main branch will be available to dogfood users.	I have clarity around when a change merged into the main branch will be available to external beta users.	I have clarity around when a change merged into the main branch will become available to all external users.	I am satisfied with the speed at which changes from main branch become available to dogfood users.	I am satisfied with the speed at which changes from main branch become available to external beta users.	I am satisfied with the speed at which changes from main branch become available to all external users.
2022Q1	3.50	3.53	2.88	3.57	3.57	3.57	3.14
2022Q2	2.83	3.95	3.28	3.73	3.67	3.67	3.83
2022Q4	2.73	3.78	3.32	3.76	3.80	3.80	3.90
Diff	-0.10	-0.17	0.03	0.02	0.14	0.14	0.07

SURVEY



			I work mostly on	We should keep faster release trains (~5 days instead of ~10).
Scale		Average by discipline	Development	3.74
5 = 5			Engineering Management	3.71
4 = 4			Product Management	3.83
3 = 3			Quality Engineering	3.75
2 = 2			Program Management	2.00
1 = 1				
Legend		Average by Layer	Product Engineering	3.71
bad			Infrastructure	3.89
good			Other	2.00
		Average by Platform	iOS	3.65
			Android	3.80
			Android|iOS	3.71
			Average	3.71

> I WOULD SAY THE PRESSURE IS LESS THAN WHAT IT WAS WITH LONGER TRAIN

A DEVELOPER

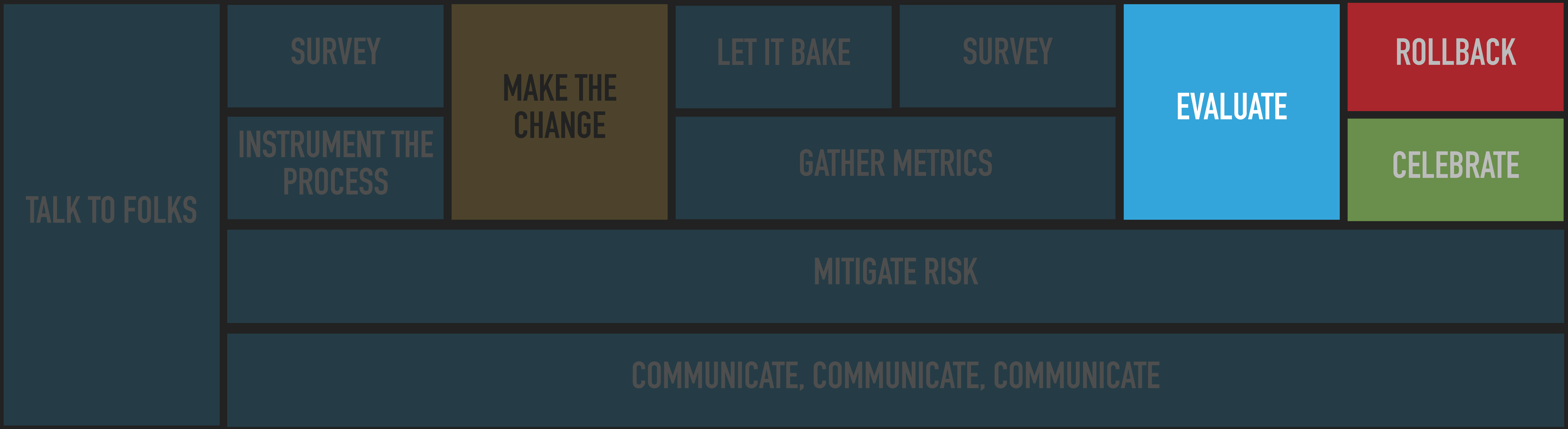
> AS LONG AS WE CONTINUE TO FEEL CONFIDENT ABOUT OUR ABILITY TO TEST THE APP BEFORE IT GOES OUT TO EXTERNAL CUSTOMERS, THIS PACE IS FANTASTIC!

A DEVELOPER

> THE 5-DAY SCHEDULE WORKS GREAT WHEN EVERYTHING GOES SMOOTHLY. IN PRACTICE, THERE ARE SEVERAL PAIN POINTS THAT MAKE THE RELEASE PROCESS HIGHER-RISK, AND LESS-FLEXIBLE…

THE RELEASE MANAGER

THE PROCESS OF CHANGE



TALK TO FOLKS

SURVEY

INSTRUMENT THE PROCESS

MAKE THE CHANGE

LET IT BAKE

SURVEY

GATHER METRICS

EVALUATE

ROLLBACK

CELEBRATE

MITIGATE RISK

COMMUNICATE, COMMUNICATE, COMMUNICATE

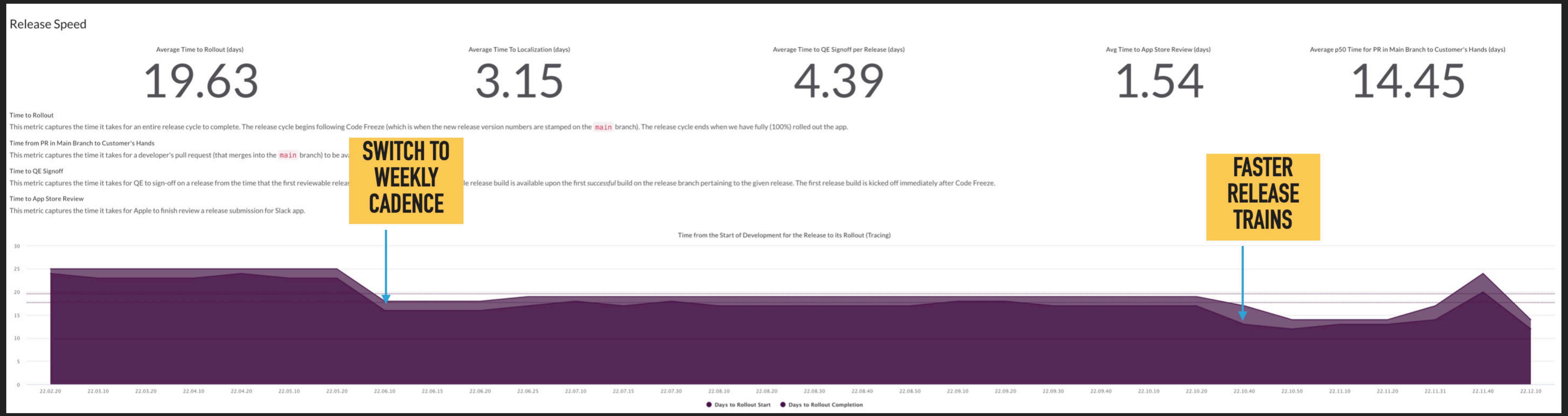
THE PROCESS OF CHANGE

EVALUATE

- Success Criteria
  - Increased velocity
  - Increased clarity
  - Reduced pressure
  - No regression in stability

EVALUATE

- Success Criteria
  - Increased velocity
  - Increased clarity
  - Reduced pressure
  - No regression in stability



Release Speed

Average Time to Rollout (days)	Average Time To Localization (days)	Average Time to QE Signoff per Release (days)	Avg Time to App Store Review (days)	Average p50 Time for PR in Main Branch to Customer's Hands (days)
19.63	3.15	4.39	1.54	14.45

Time to Rollout

This metric captures the time it takes for an entire release cycle to complete. The release cycle begins following Code Freeze (which is when the new release version numbers are stamped on the `main` branch). The release cycle ends when we have fully (100%) rolled out the app.

Time from PR in Main Branch to Customer's Hands

This metric captures the time it takes for a developer's pull request (that merges into the `main` branch) to be av

Time to QE Signoff

This metric captures the time it takes for QE to sign-off on a release from the time that the first reviewable releas le release build is available upon the first *successful* build on the release branch pertaining to the given release. The first release build is kicked off immediately after Code Freeze.

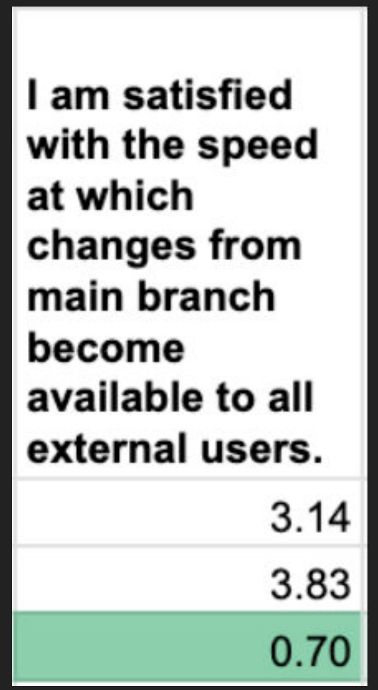
Time to App Store Review

This metric captures the time it takes for Apple to finish review a release submission for Slack app.

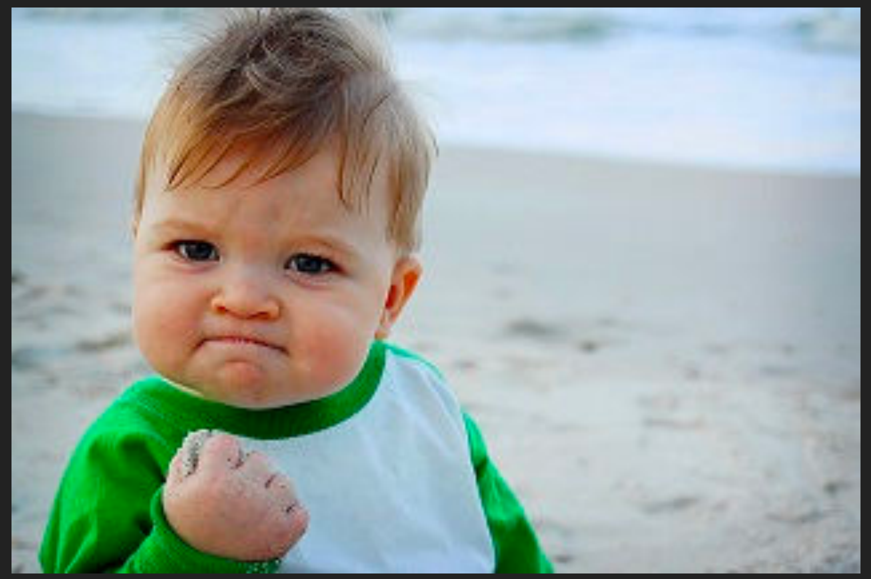
SWITCH TO WEEKLY CADENCE

FASTER RELEASE TRAINS

Time from the Start of Development for the Release to its Rollout (Tracing)
30
25
20
15
10
5
0
22.02.20
22.03.10
22.03.20
22.04.10
22.04.20
22.05.10
22.05.20
22.06.10
22.06.15
22.06.20
22.06.25
22.07.10
22.07.15
22.07.30
22.08.10
22.08.20
22.08.30
22.08.40
22.08.50
22.09.10
22.09.20
22.09.30
22.09.40
22.10.10
22.10.20
22.10.40
22.10.50
22.11.10
22.11.20
22.11.31
22.11.40
22.12.10
Days to Rollout Start
Days to Rollout Completion



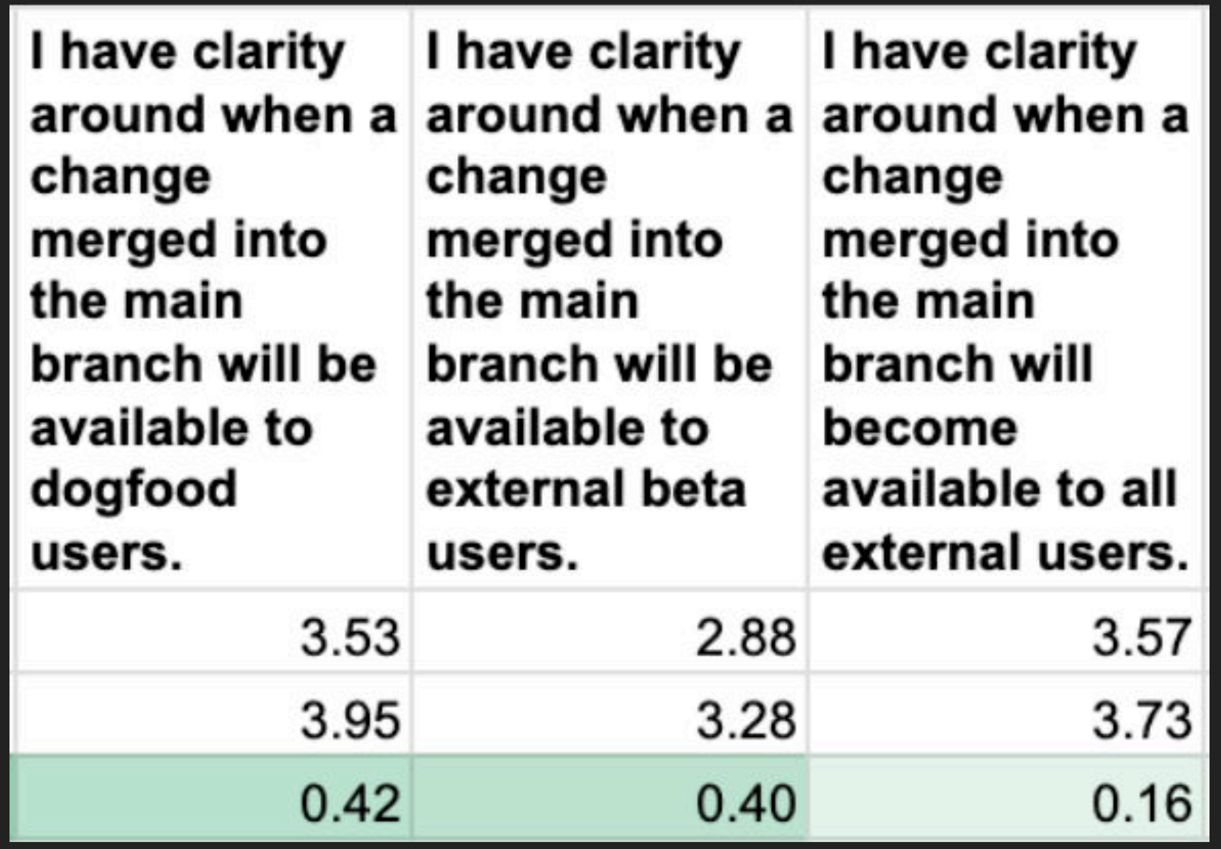
I am satisfied with the speed at which changes from main branch become available to all external users.
3.14
3.83
0.70



THE PROCESS OF CHANGE

EVALUATE

- Success Criteria
  - Increased velocity
  - Increased clarity
  - Reduced pressure
  - No regression in stability

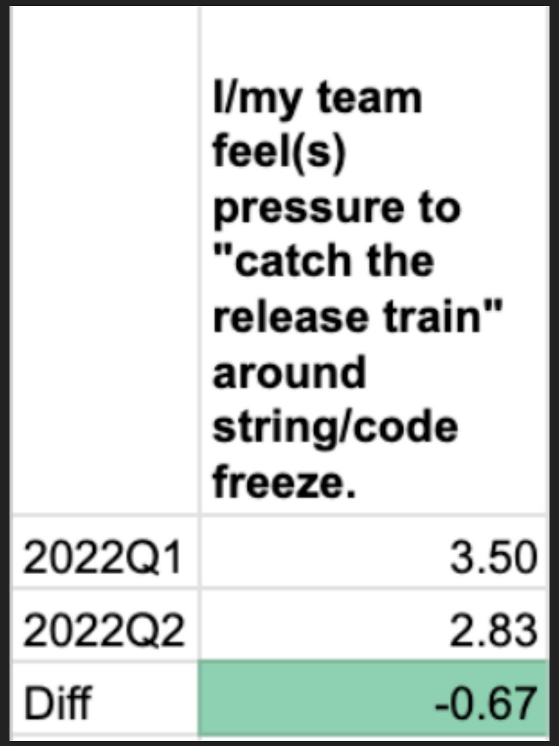


I have clarity around when a change merged into the main branch will be available to dogfood users.	I have clarity around when a change merged into the main branch will be available to external beta users.	I have clarity around when a change merged into the main branch will become available to all external users.
3.53	2.88	3.57
3.95	3.28	3.73
0.42	0.40	0.16

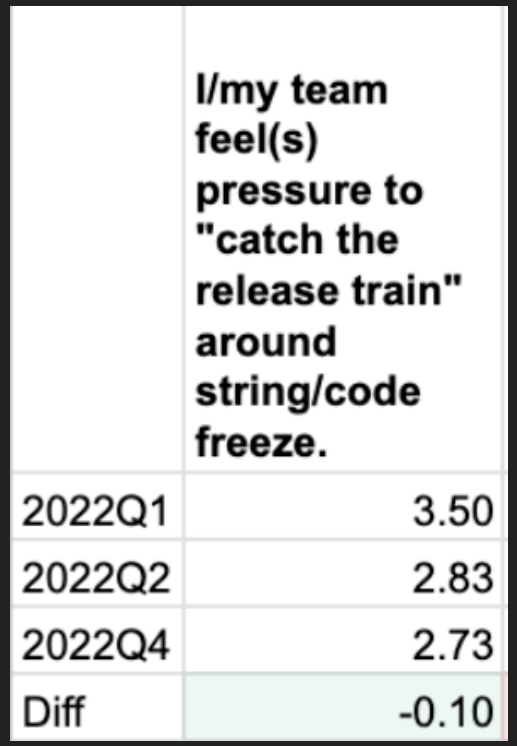
THE PROCESS OF CHANGE

EVALUATE

- Success Criteria
  - Increased velocity
  - Increased clarity
  - Reduced pressure
  - No regression in stability



	I/my team feel(s) pressure to "catch the release train" around string/code freeze.
2022Q1	3.50
2022Q2	2.83
Diff	-0.67

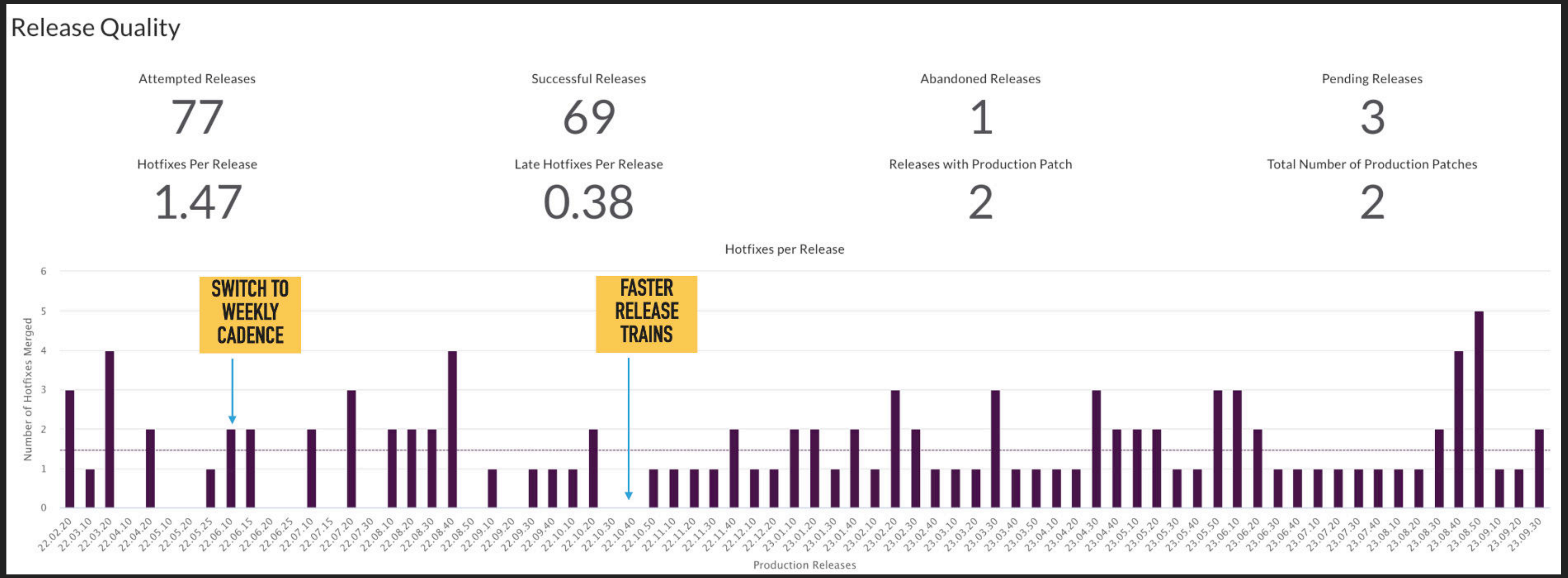


	I/my team feel(s) pressure to "catch the release train" around string/code freeze.
2022Q1	3.50
2022Q2	2.83
2022Q4	2.73
Diff	-0.10

THE PROCESS OF CHANGE

EVALUATE

- Success Criteria
  - Increased velocity
  - Increased clarity
  - Reduced pressure
  - No regression in stability



Release Quality

Attempted Releases
77

Successful Releases
69

Abandoned Releases
1

Pending Releases
3

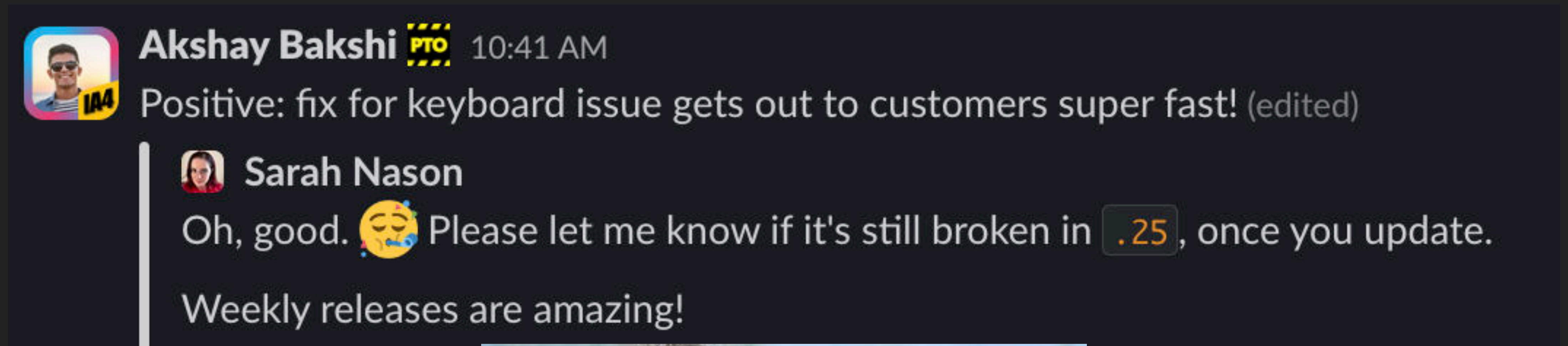
Hotfixes Per Release
1.47

Late Hotfixes Per Release
0.38

Releases with Production Patch
2

Total Number of Production Patches
2

EVALUATE

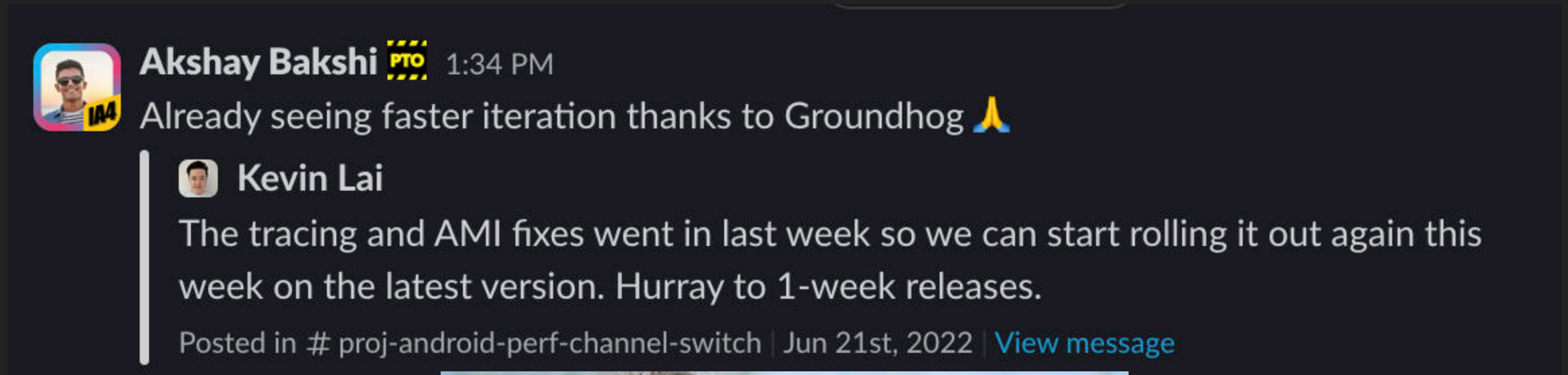


Akshay Bakshi PTO 10:41 AM

Positive: fix for keyboard issue gets out to customers super fast! (edited)

> **Sarah Nason**
>
> Oh, good. 🥹 Please let me know if it's still broken in `.25`, once you update.
>
> Weekly releases are amazing!
>
> From a thread in # io

reply

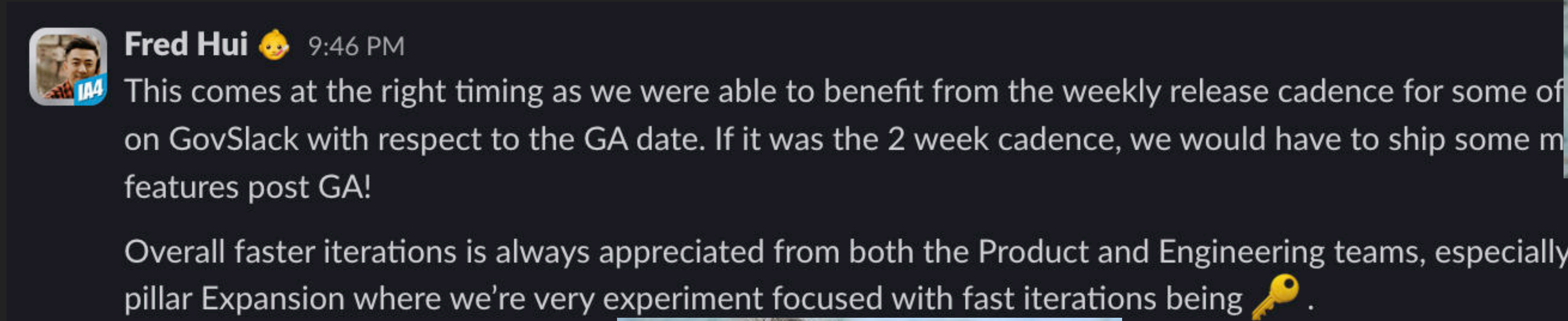


Akshay Bakshi PTO 1:34 PM

Already seeing faster iteration thanks to Groundhog 🙏

> **Kevin Lai**
>
> The tracing and AMI fixes went in last week so we can start rolling it out again this week on the latest version. Hurray to 1-week releases.
>
> Posted in # proj-android-perf-channel-switch | Jun 21st, 2022 | View message

4 1 1



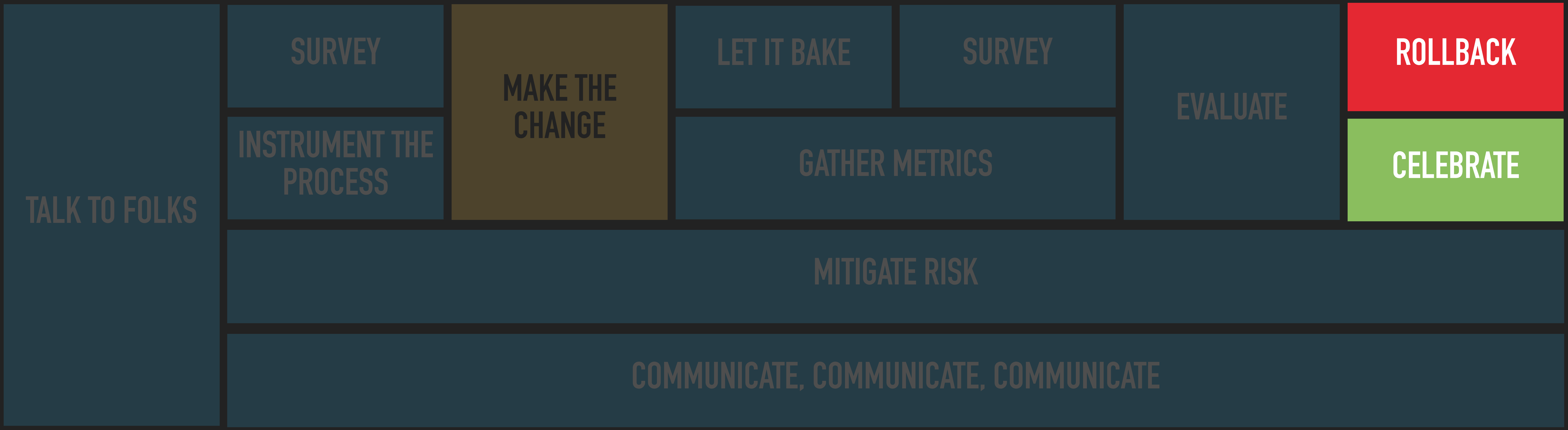
Fred Hui 9:46 PM

This comes at the right timing as we were able to benefit from the weekly release cadence for some of on GovSlack with respect to the GA date. If it was the 2 week cadence, we would have to ship some m features post GA!

Overall faster iterations is always appreciated from both the Product and Engineering teams, especially in my pillar Expansion where we're very experiment focused with fast iterations being 🔑 .

2 1

THE PROCESS OF CHANGE



TALK TO FOLKS

SURVEY

INSTRUMENT THE PROCESS

MAKE THE CHANGE

LET IT BAKE

SURVEY

GATHER METRICS

EVALUATE

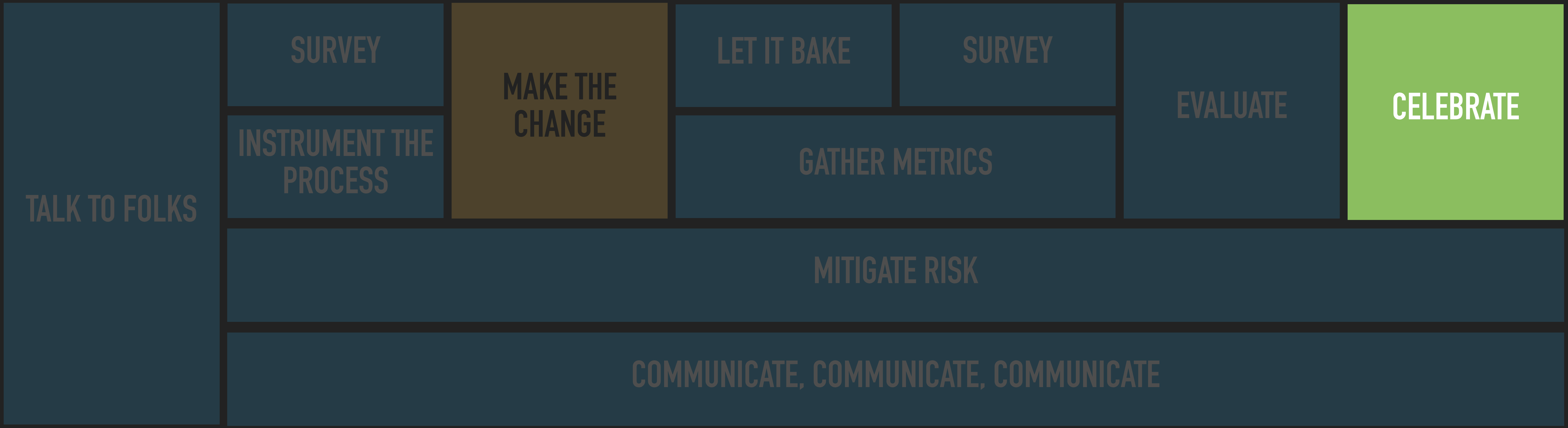
ROLLBACK

CELEBRATE

MITIGATE RISK

COMMUNICATE, COMMUNICATE, COMMUNICATE

THE PROCESS OF CHANGE



TALK TO FOLKS

SURVEY

INSTRUMENT THE PROCESS

MAKE THE CHANGE

LET IT BAKE

SURVEY

GATHER METRICS

EVALUATE

CELEBRATE

MITIGATE RISK

COMMUNICATE, COMMUNICATE, COMMUNICATE

FUTURE POSSIBILITIES

Software delivery performance metric	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per week and once per month	Between once per month and once every 6 months	Fewer than once per six months
Lead time for changes For the primary application or service you work on, w is your lead time for changes (i.e., how long does it t to go from code committed to code successfully ru in production)?	Less than	Between day and week	Between one month and six months	More than six months
Time to restore service For the primary application or service you work on, long does it generally take to restore service when a service incident or a defect that impacts users occu (e.g., unplanned outage or service impairment)?		han ay	Between one day and one week	More than six months
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0%-15%	16%-30%	16%-30%	16%-30%

OPPORTUNITIES

- Iterate with “closer” audiences
    - Early prototypes to key stakeholders
    - Internal users
    - External beta users

- Shift Left
    - Reduce risk of hotfixes
    - Get to releasing a green build from main

- AI

FINAL THOUGHTS

THE END

FINAL THOUGHTS

- Change triggers fear
- “Fear is the mind killer”
- It’s easy to get into a comfort zone
- With the right approach we can reach a new level of productivity

Which, if you have read book 4 of the dune series, is something that we probably don’t want to do. But that philosophical talk for next year. If AI hasn’t become sentient and taken over by then. See you all in the future.