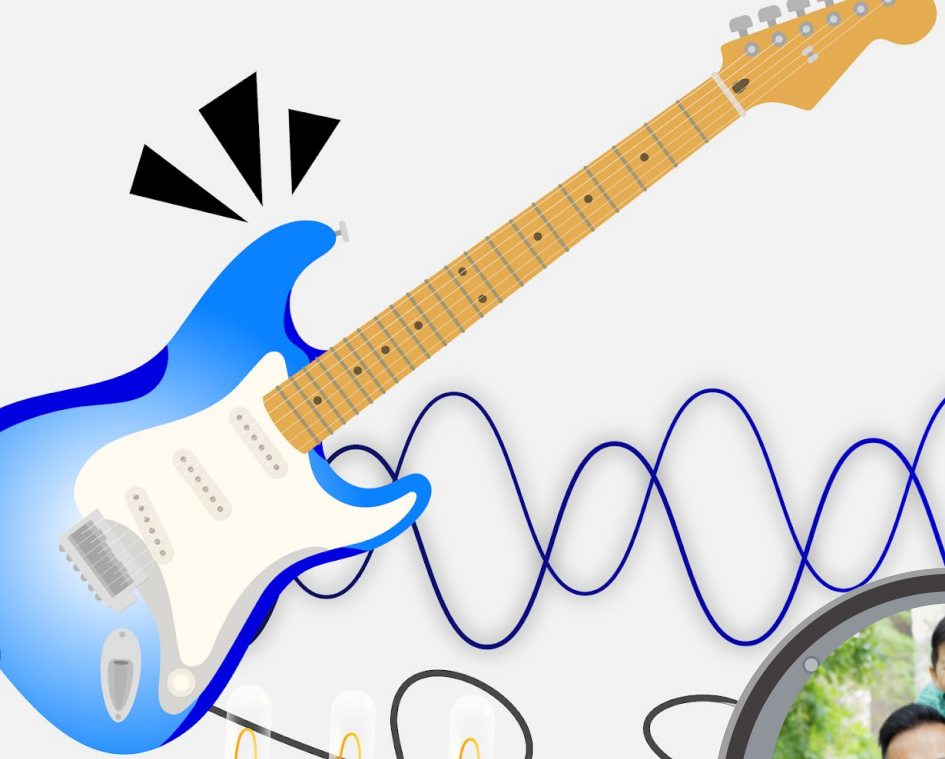
PNSQC

OCTOBER 9-11 2023

AMP IT UP:

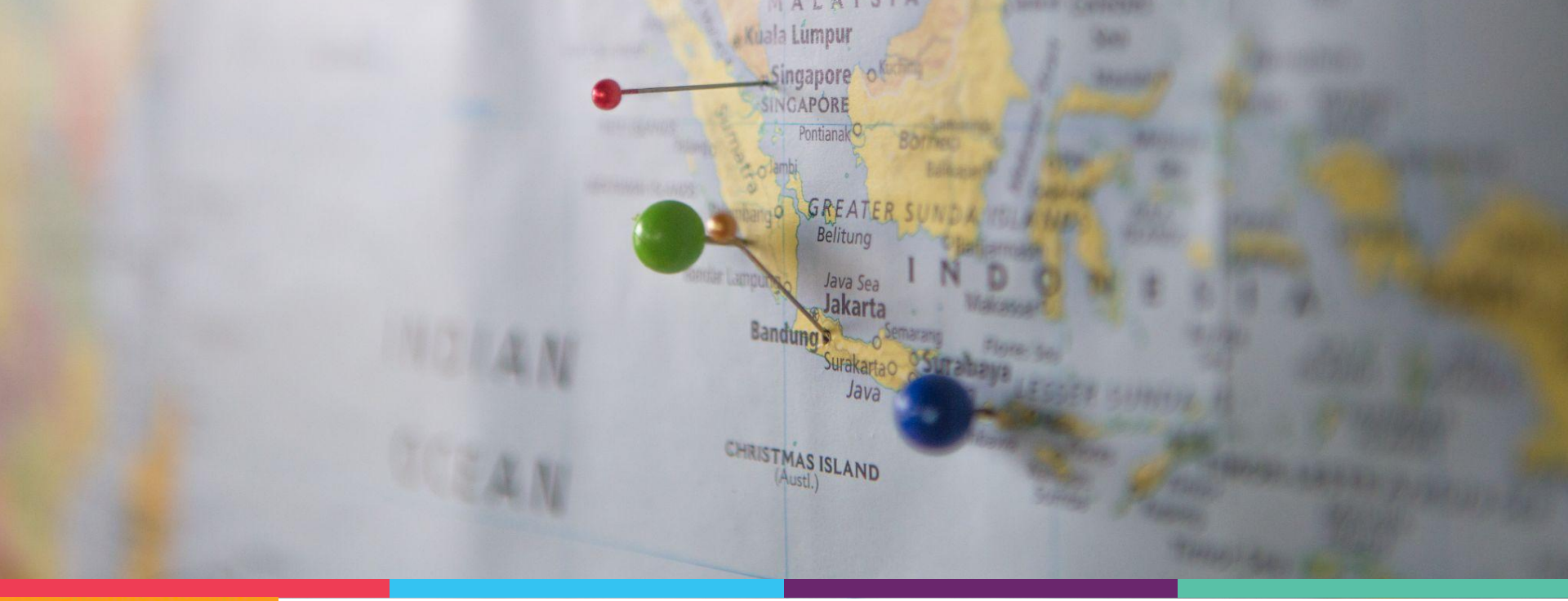
TRANSFORMING QUALITY



PNSQC.ORG

Jeff Sing

QA STRATEGY



41ST ANNUAL
PACIFIC NW SOFTWARE QUALITY CONFERENCE
OCTOBER 9-11, 2023

Establishing your Quality Roadmap through Quarterly Service Delivery Reviews

Jeff Sing | *Iterable*

Jeff Sing

Director of Engineering - Quality & Operations
@Iterable

jeff.sing@iterable.com

https://www.linkedin.com/in/jeffsing/

17 YEARS OF QA

Skylight Healthcare (Acquired by Getwell Networks)

- Sr Software QA Engineer testing hardware + software + builds
- Managed deploys and Tier 3 support

iOvation (Acquired by TransUnion)

- Software Automation Engineer developing Ruby automation to test microservices
- Full stack testing: BE/FE + managed deploys through Puppet/Jenkins/Rundeck

Optimizely (Acquired by Episerver)

- Chief of Staff (Office of the CTO) & Engineering Manager - Quality & Operations
- Built and led the QA Team
- Led the TPM org and supported the Engineering Leadership Organization

Iterable

- Director of Engineering, Quality & Operations
- Built and led the QA Team
- Built and led the Engineering Operations Team

THE QUALITY PROGRAM

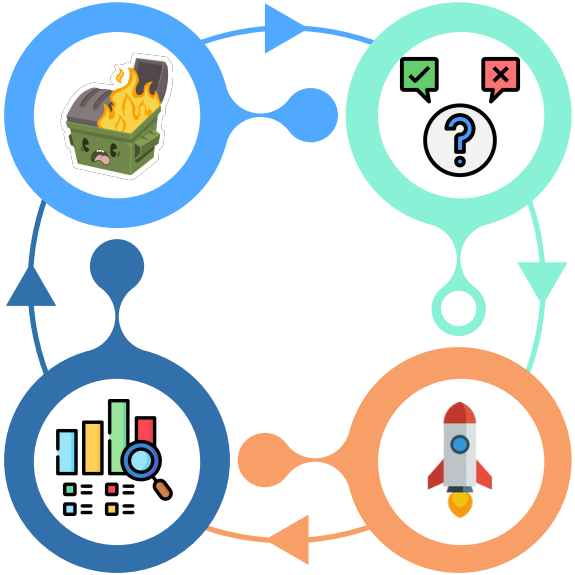
HOW DO YOU DETERMINE WHAT TO WORK ON?

Identify Problem

Where is our quality bad? What is hurting our business?

Define Success

What does good look like? How do we tell?



Review Data

Collect results. Run the reviews and retrospectives to validate our assumptions.

Launch Initiatives

Execute technical solutions & launch quality programs identified to get success (QE or Partners)


WHY IS THIS CHALLENGING?

Example: Testing seems so simple!

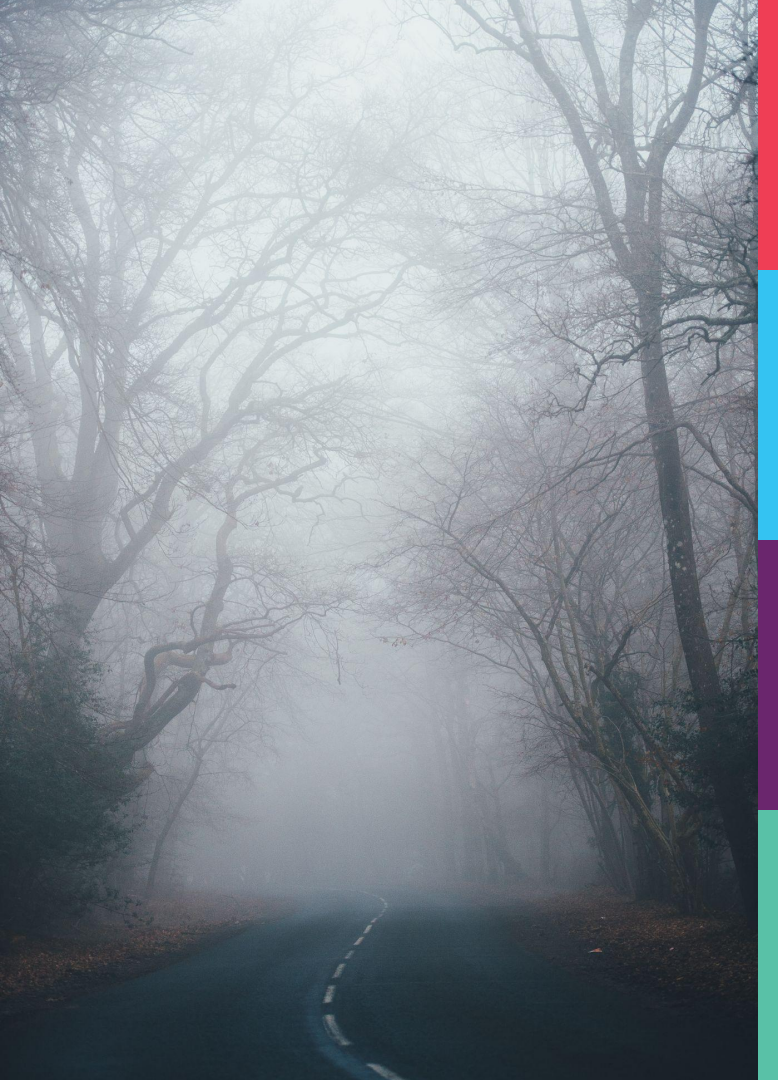
Incidents have gone up, so your CTO wants you to s*moke test every single page* in your web app. Why not?

Example: EM equate QE to TESTING only...

EM only need QA in the end of the SDLC to complete the testing initiatives and aren't bringing them in earlier or empowering them to come in earlier.

Example: Use Vendor X to achieve Quality goals

Director of Engineering heard a pitch that this vendor can solve all our quality problems cuz they have revolutionized software testing. They can easily optimize and scale our testing by 300%!



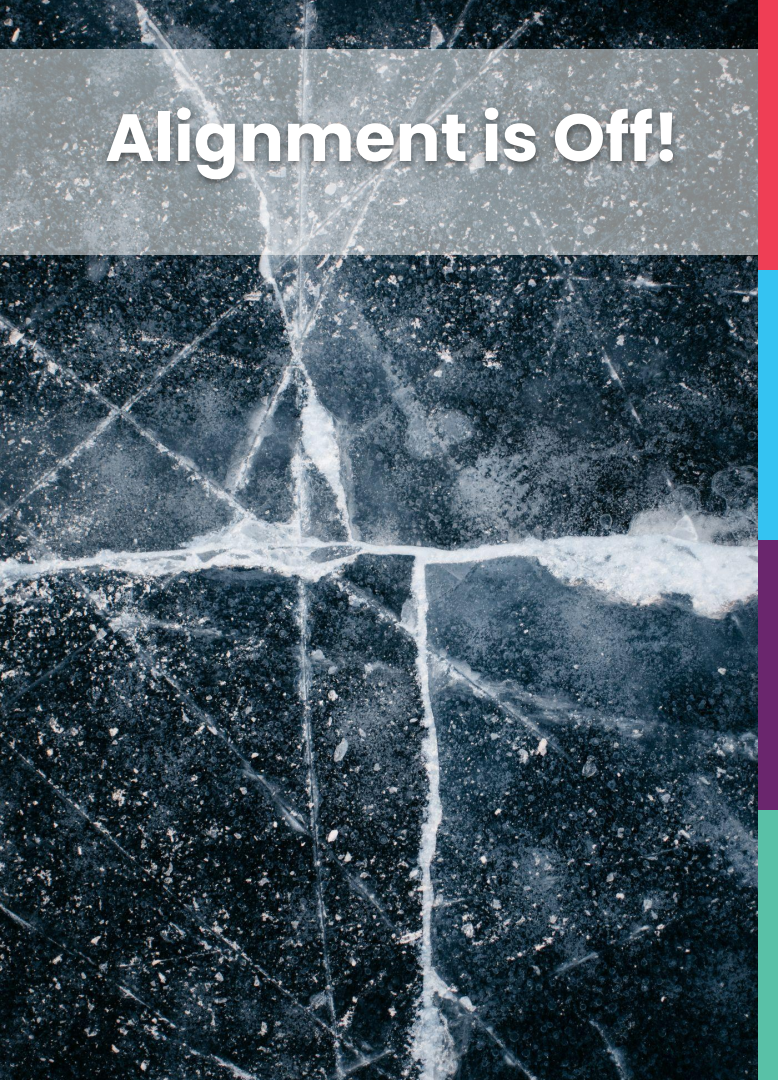
Success is Not Clear

How do we measure Quality Success?

Does our metrics:

- Align with our high level OKRs?
- Determine if engineers are releasing better code?
- Actually measure customers happiness?

Alignment is Off!



Example: What does QA do if they don't write tests?

Why can't QA write all the automation? Developers are to busy shipping code.

Example: Teams don't follow the SDLC

Engineering skipping steps in the SDLC and not following the right process leading to major mishaps.

Example: What happened to shift left?

Engineering does not fund shift left work and depends too much on the latter testing process catching things late in the SDLC.

How do you communicate your strategy?

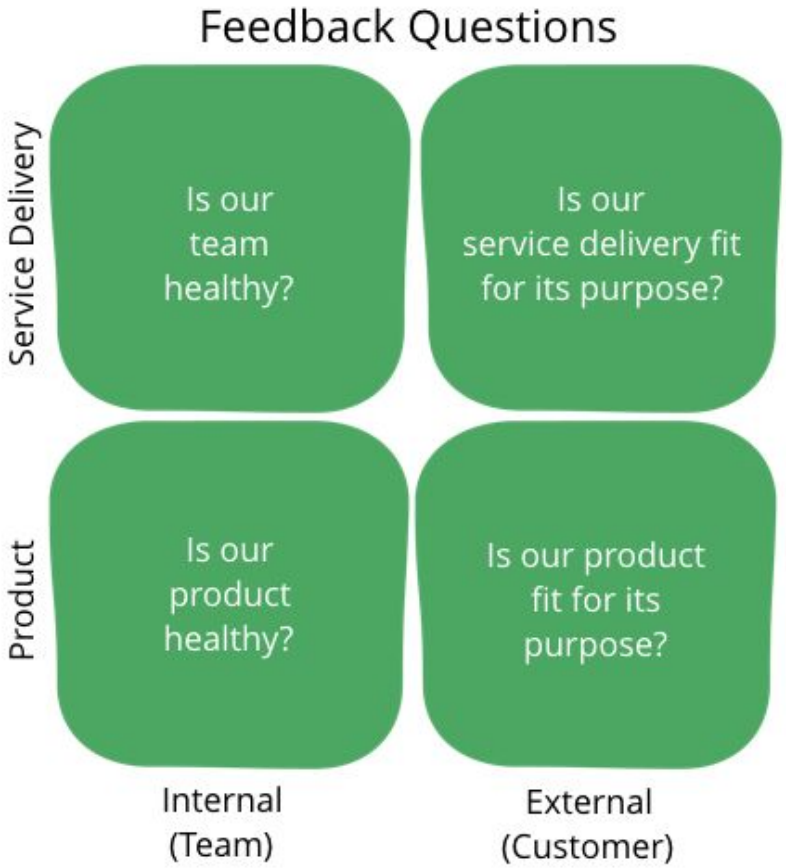
You know what to do, and you have a strategy. But now what?

1. How do you communicate this to your peers and partner organizations like customer support to align with you?
2. How do you introduce new initiatives that potentially could clash in prioritization?
3. How do you get buy in from other organizations to execute your quality strategy?

QUALITY SERVICE DELIVERY REVIEW

WHAT IS A QUALITY SERVICE DELIVERY REVIEW?

- **Feedback loop** that facilitates a quantitatively-oriented discussion between a customer and delivery team about the **fitness for purpose** of its service delivery
- Understand what **mechanisms** we use to be able to continuously measure how well we are fulfilling the *customer's reason for choosing us.*
- **Drive discussion** and help team agree upon actions to take to improve the system's capability.



CYCLES OF A SERVICE DELIVERY REVIEW

Iterate

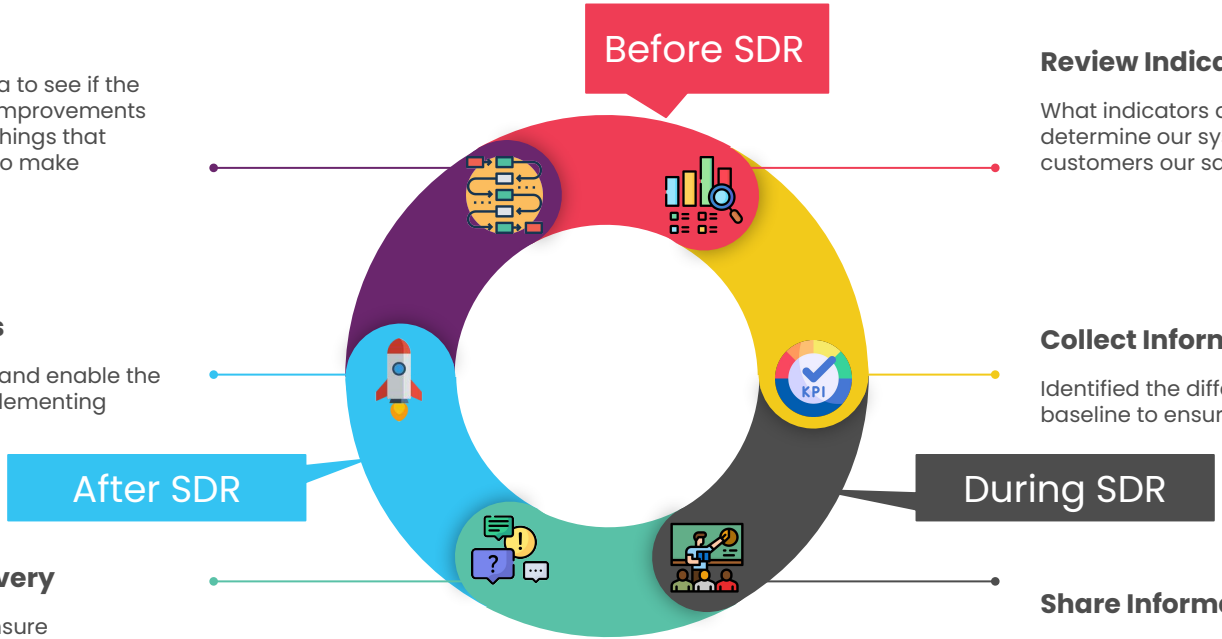
Constantly review the data to see if the initiatives are driving the improvements you have proposed. Flag things that aren't working and be ok to make changes on the fly.

Implement Changes

Create the high level OKR and enable the organization to begin implementing changes as planned.

Discuss Service Delivery

Utilizing data collected, ensure participants are discussing issues and aligning on ownership of problem areas to improve quality.



Review Indicators to Monitor

What indicators do we want to monitor to determine our system quality, and if our customers our satisfied with our services?

Collect Information

Identified the different sources and baseline to ensure trustworthiness.

Share Information

Ensure the right stakeholders and partners have visibility into your data.

SERVICE DELIVERY FLOW

Recap	Quality KPI	State of QA	Discussion
• How are we doing in regards to our in flight quality initiatives • What is still not finished and should be paid attention to	• Quick definition of each KPI + Business Impact • Recap data for current quarter • Highlight trends • Provide data on outliers or callouts	• Visibility into resources and what they are working on • Introduce upcoming Roadmap, how each initiative is tied to certain outcomes + areas of partnership • Highlight areas of risk	• Highlight big quality initiatives that matter, showcase the business impact, and present the problems itemized • Enforce discussion amongst group (pro's vs con's), get buy in.

BEFORE YOU RUN YOUR QUALITY SERVICE DELIVERY REVIEW

DATA TO BASELINE

WARNING: METRICS WITHOUT CONTEXT

Metrics by themselves don’t give the full picture if we actually are hitting our quality target.

94%

Test Coverage

500+

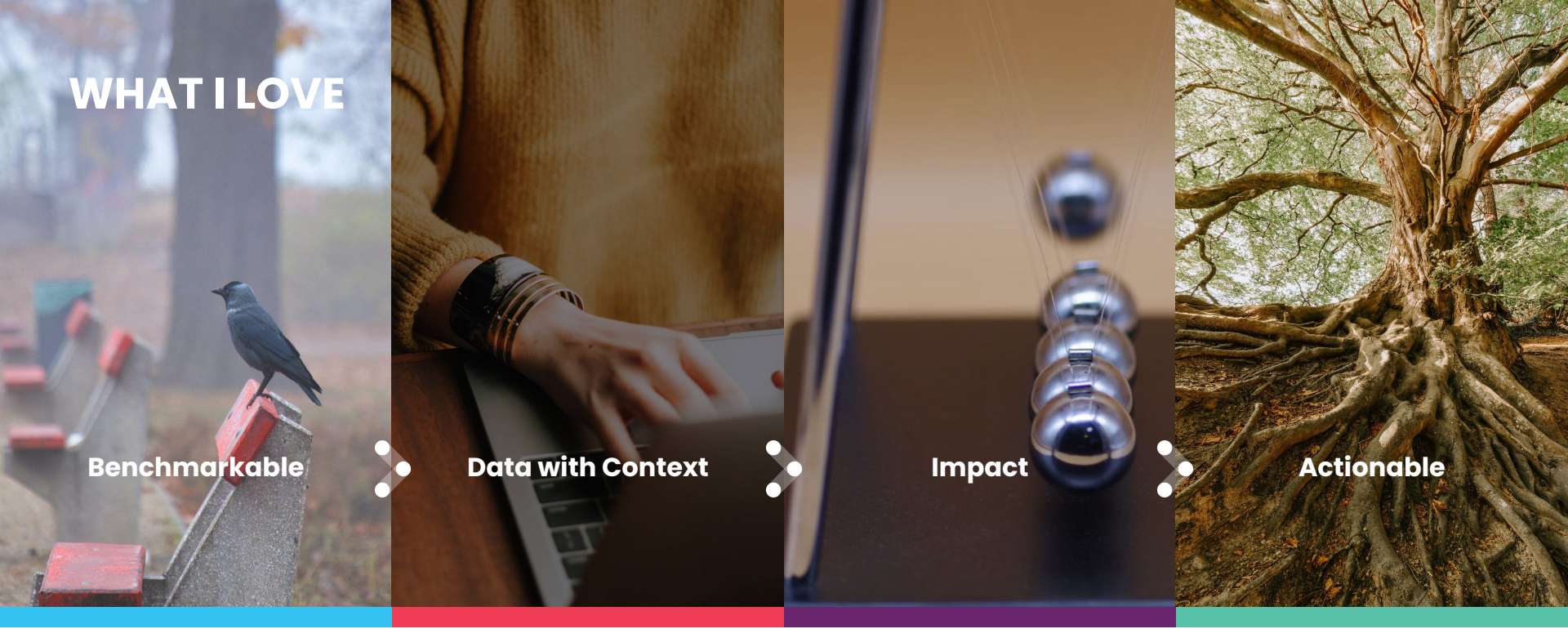
Test Cases Created

46

Bugs Reported

300

E2E Test Run



DORA
Change Failure Rate

Velocity
# of Bugs / # of PR

High Sev
Customer Reported

Root Cause
Code vs Config vs System

BENCHMARKABLE DATA

How do we compare to our peers in the industry?

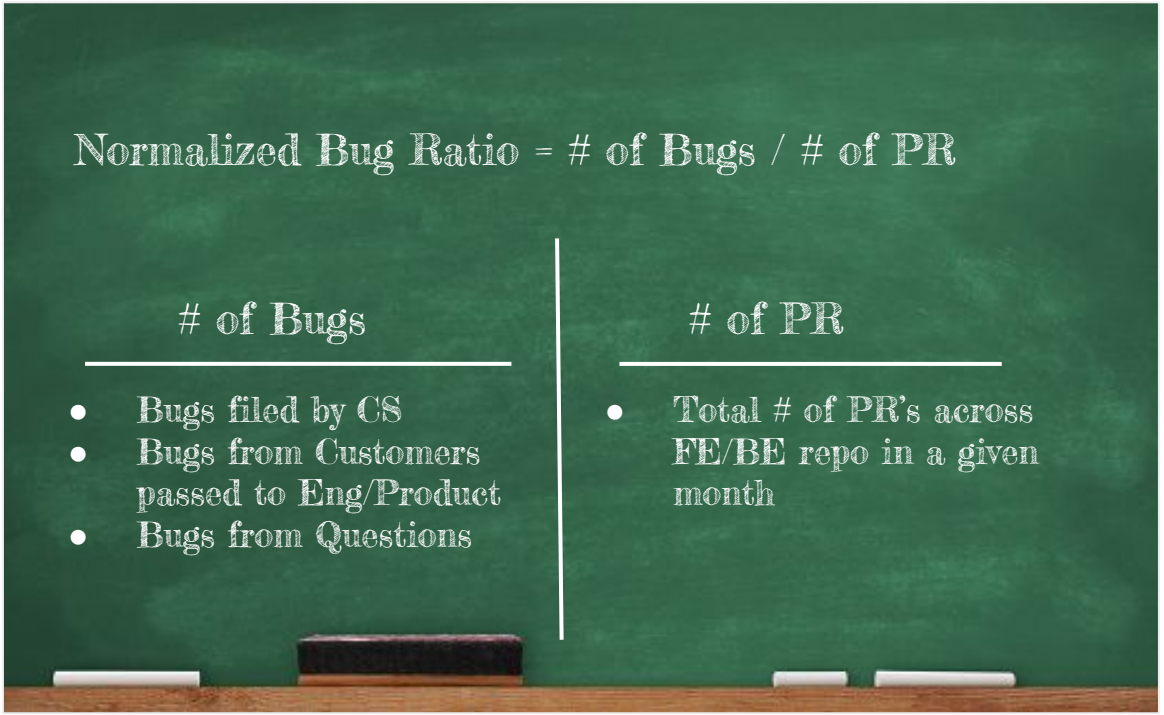
Helps me establish a target + determine if there is something systematically wrong or high risks.

Software delivery performance metric	Low	Medium	High
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?	Between once per month and once every 6 months	Between once per week and once per month	On-demand (multiple deploys per day)
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)?	Between one month and six months	Between one week and one month	Between one day and one week
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)?	Between one week and one month	Between one day and one week	Less than one day
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)?	46%-60%	16%-30%	0%-15%

DATA WITH CONTEXT

Does the data make sense?

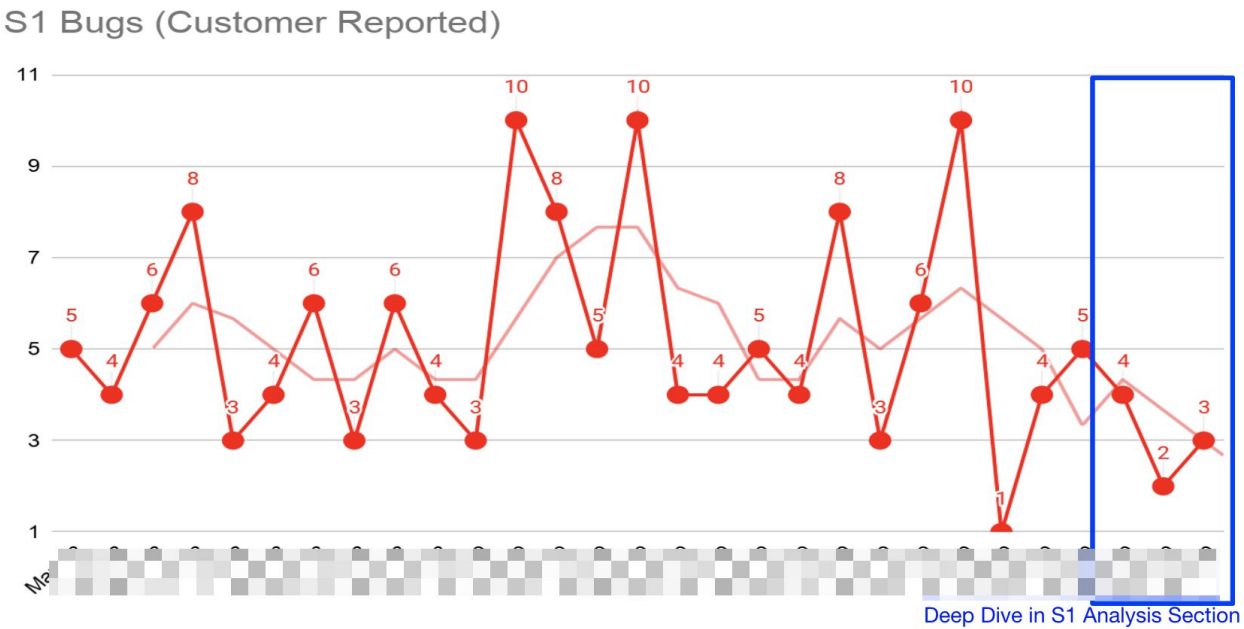
Ability to measure feature quality based on developer velocity. Ultimate goal would be to see a smaller % as that would mean we are shipping with high velocity and high quality.



DATA ON IMPACT

How often are we failing our customers?

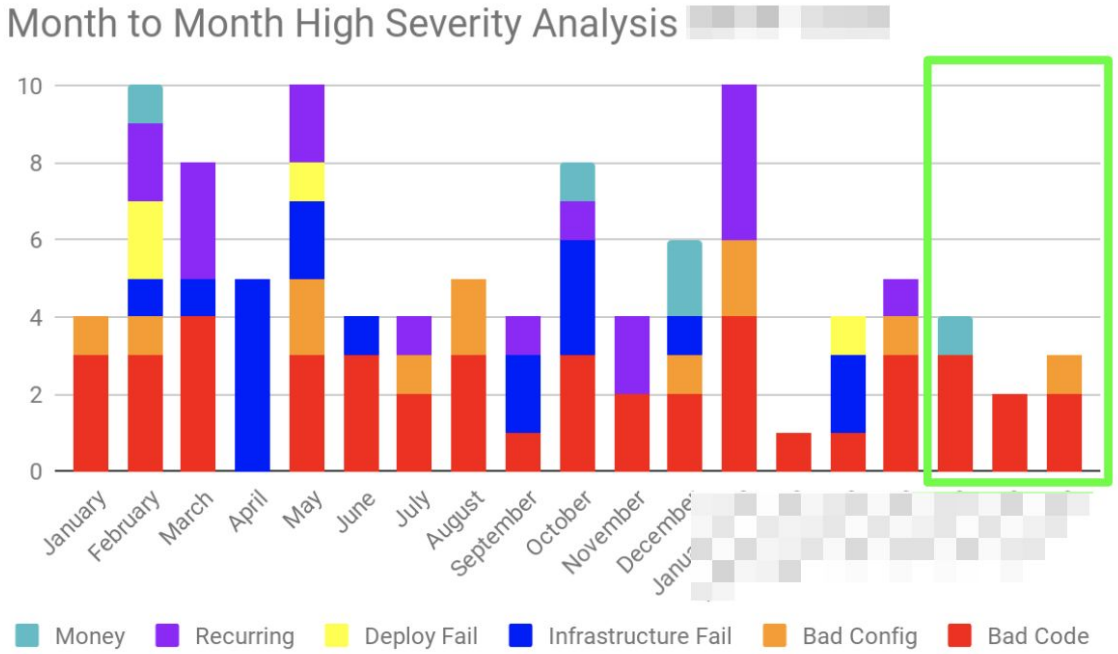
Quality is ultimately determined by our customers. How many times have we failed them?



DATA ON ROOT CAUSE

What are we failing on?

Is there a commonality on what we are failing on? Are there trends we should be aware of?

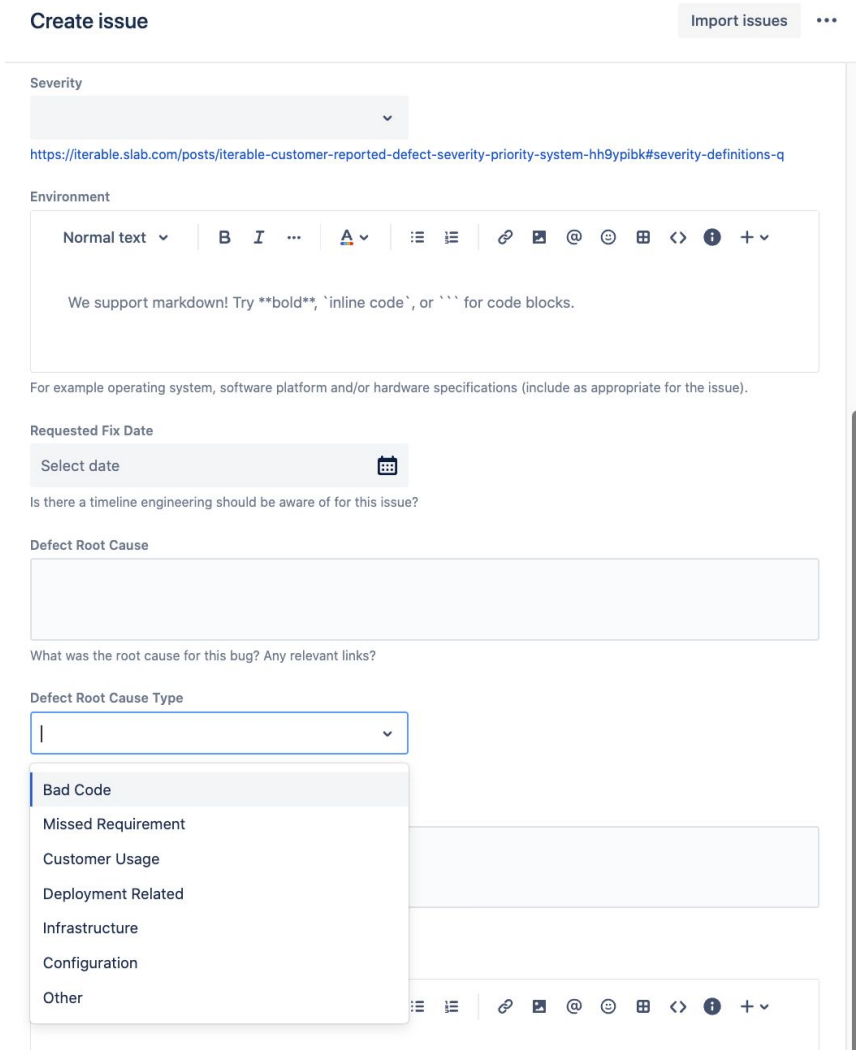



COLLECTING DATA

Jira, your best friend?

Mandatory collection of data. Examples:

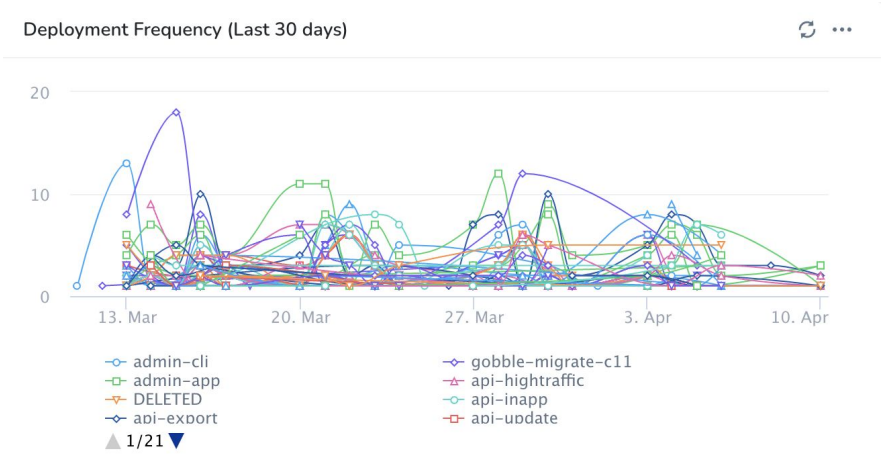
- Root Cause
- Severity
- Prevention
- SLA



DATA ON DEPLOYMENTS

Collecting the Velocity of Output

Having this info allows you to give context over the quality of what you are deploying. The challenge is collecting this information, as the maturity of your deployment pipeline varies from company to company.



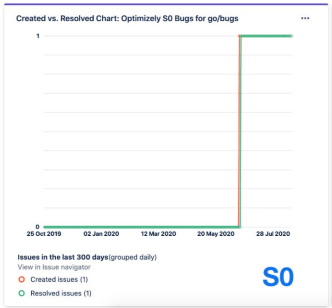
SHOWCASE QUALITY KPI TRENDS Q2Q/M2M

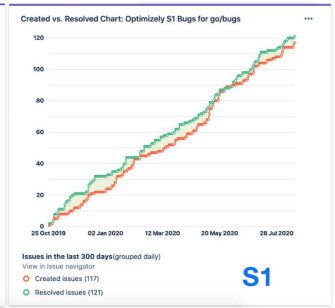
Jira Defects S0 - S3

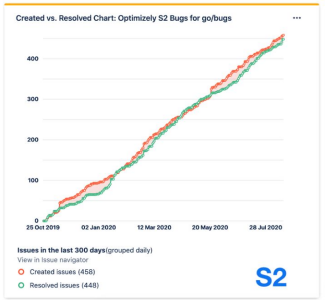
Trends over the last 300 days

[Dashboard here]

Sev	Total	Per Eng (72)
S0	1 ↑	0
S1	117 ↓	2
S2	458 ↓	6
S3	228 ↓	3







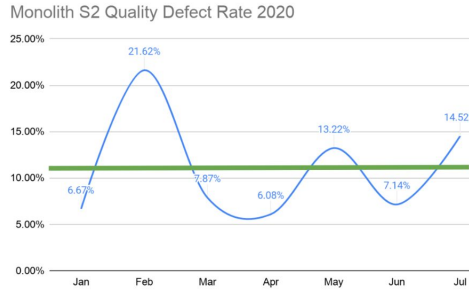
Monolith Quality Trend

[2020 TSE Calculation Sheet]

2.5%

	Jan	Feb	Mar	Apr	May	Jun	Jul
	10.00%	0.00%	2.36%	1.66%	2.48%	2.04%	1.61%

11%



S2 Defect Rate	Jan	Feb	Mar	Apr	May	Jun	
Monolith	6.67%	21.62%	7.87%	6.08%	13.22%	7.14%	14

DURING THE QUALITY SERVICE DELIVERY REVIEW

DATA TO ACTION

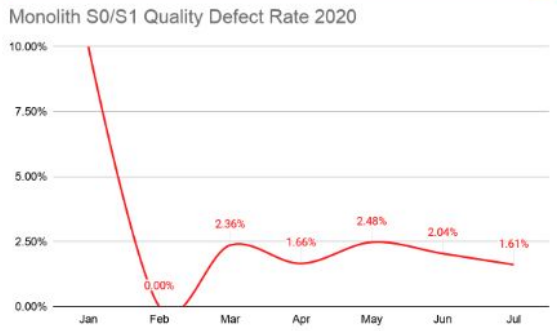
SHOWCASE KPI & HYPOTHESIS OF ROOT CAUSE

Analysis of Trends

What is baseline healthy? Was there some event that caused a significant change? Is there something I should double click into?

$$TT \div C = \textit{Defect Rate}$$

TT - **TOTAL TICKETS** (Customer Reported)
C - Number of **Commits** Released in Weekly Manifest

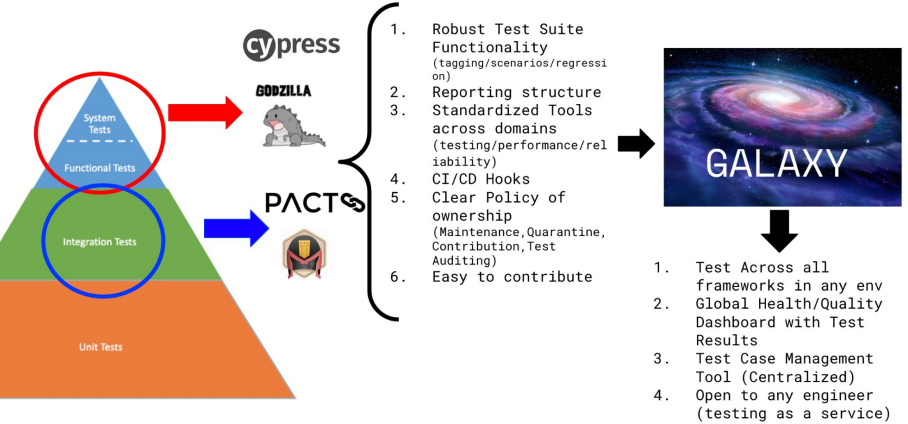


S1 Defect Rate	Jan	Feb	Mar	Apr	May	Jun	Jul
Monolith	10.00%	0.00%	2.36%	1.66%	2.48%	2.04%	1.61%

S2 Defect Rate	Jan	Feb	Mar	Apr	May	Jun	Jul
Monolith	6.67%	21.62%	7.87%	6.08%	13.22%	7.14%	14.52%

STATE OF QA EXAMPLES: RESOURCE ALLOCATION

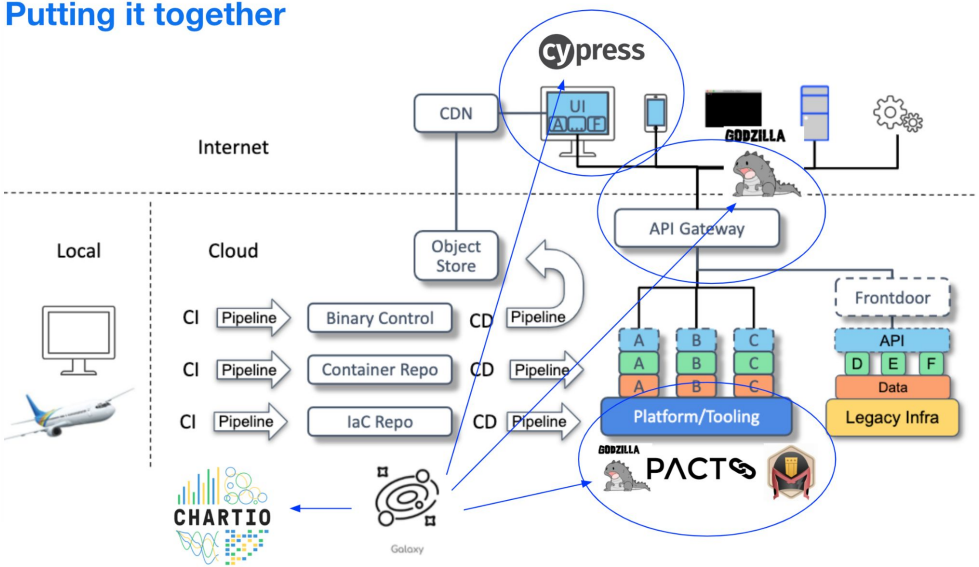
Putting it together



1. Robust Test Suite Functionality (tagging/scenarios/regression)
2. Reporting structure
3. Standardized Tools across domains (testing/performance/reliability)
4. CI/CD Hooks
5. Clear Policy of ownership (Maintenance, Quarantine, Contribution, Test Auditing)
6. Easy to contribute

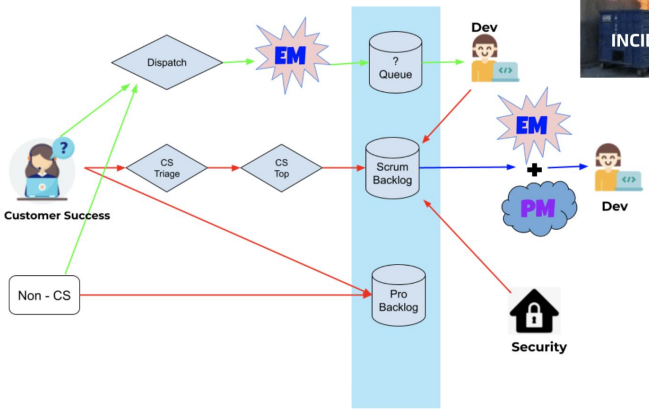
1. Test Across all frameworks in any env
2. Global Health/Quality Dashboard with Test Results
3. Test Case Management Tool (Centralized)
4. Open to any engineer (testing as a service)

Putting it together

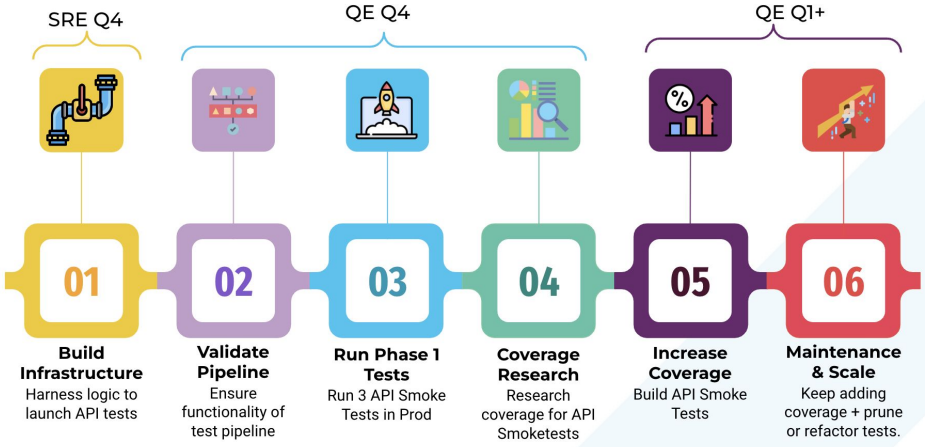


STATE OF QA EXAMPLES: QUALITY INITIATIVES

What systems are we managing?



API SMOKE TEST IN PRODUCTION (MILESTONES)



DISCUSSION

SET THE STAGE

DATA AND IMPACT

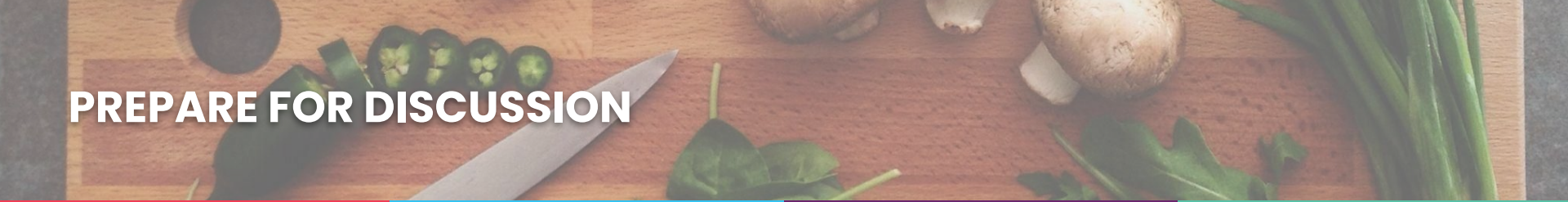
- Actual data gathered (try to include source)
- What is the business impact?
- Why is this more important than other projects in flight?

Process Improvement: Shared feature reliability

Problem statement: Shared features has higher risk introduce bug in the production.

Examples: (From Jan 10 - present)

1. Incident-1261 (S3) -
2. Incident-1265 (S3) -
3. Incident-1272 (S3) -
4. Incident-1274 (S3) -
5. Three Bugs found by Dev/QE on journey and



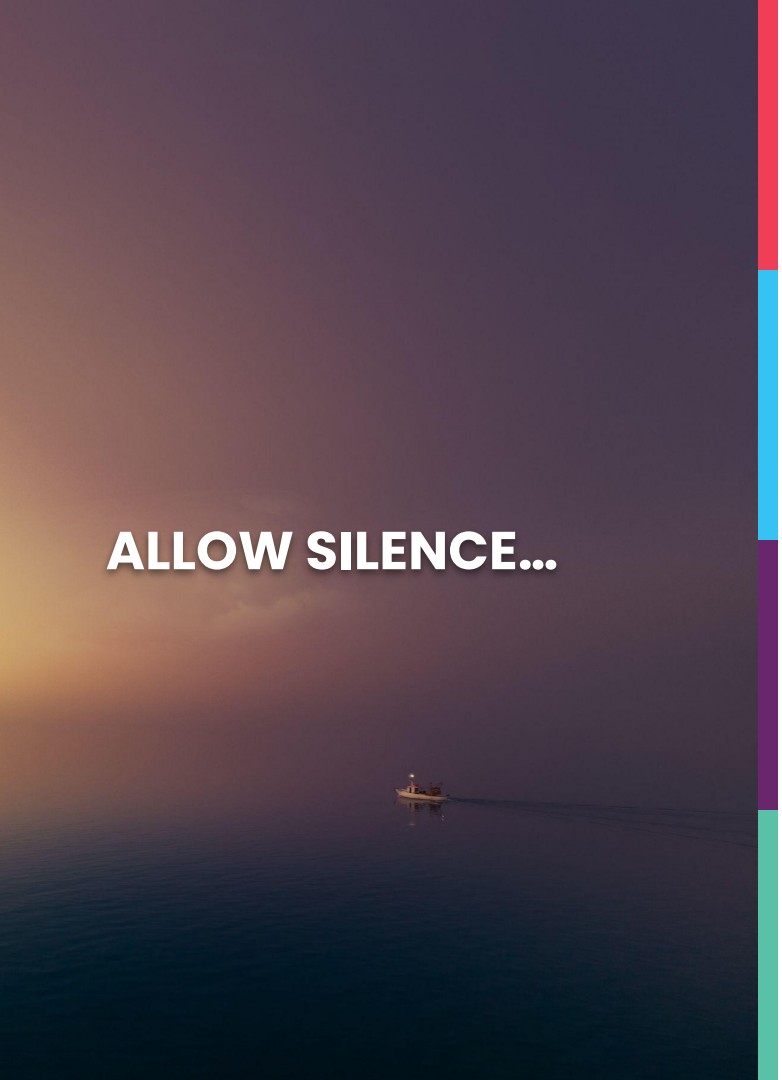
PREPARE FOR DISCUSSION

NEXT STEPS

- Lead with what you think success looks like
- Ensure the GOALS of what you are trying to solve during discussion is clear

BUG BACKLOG CLEANUP

IMPACT	NEXT STEPS
Backlog reduced from ~213 to ~101 • 52.5% reduction...yowza! • consolidation of duplicates • appropriate team assignment • clearer picture of real bugs	Discuss: • Team by team clean up? Who leads this effort? EM? QE? • SLA to auto close bugs after X days? ○ Special label so we can track closed to bankruptcy?



Silence isn't bad!

People often need time to think and internalize before they are open for dialogue. Also it gives people time to reflect on your content.

Don't Rush the Silence!

Give it a long pause, since people tend to wait for someone to go first. You really want your participants to be engaged.

Facilitate the Silence

If there is really no engagement, it's ok to prod a bit. I've used:

- "I'd love to hear some ideas from people who haven't spoken yet in this discussion."
- "Can I get someone to share your initial thoughts on this? Who is willing to start?"
- "Does someone want to post in comments, I can read it out loud?"
- "Is what I brought up confusing? Can I clarify in any ways?"

Keep on Target

Remember, you are the facilitator, don't let the meeting get off track. Course correct back onto the topic at hand.

Budget Your Time Correctly

Think of building an action plan, make sure you have enough time for each stage of this created. Use the right tactics to speed things up, or know when to slow down to deep dive in on something.

End with an Action Plan

Do not end unless there is a clear next step.

- Who are the new stakeholders and what are they responsible for?
- What is the milestones and expectations for delivery?
- Did you tie back the takeaways into solid initiatives for your Quality Roadmap?
- Can you align this into your OKRs? Does your OKR align with your bosses and departments? Make sure this is documented.

AFTER THE QUALITY SERVICE DELIVERY REVIEW

WHY WE ITERATE

FAILED INITIATIVE

- Break apart any small wins and ensure they don't die.
- Communicate next steps. Ensure you have a backlog of other projects to pivot to (based off impact).
- Create a retrospective over what went poorly, and how we can improve.
- Include learnings in your future approaches (Don't quit!)

BAD HYPOTHESIS

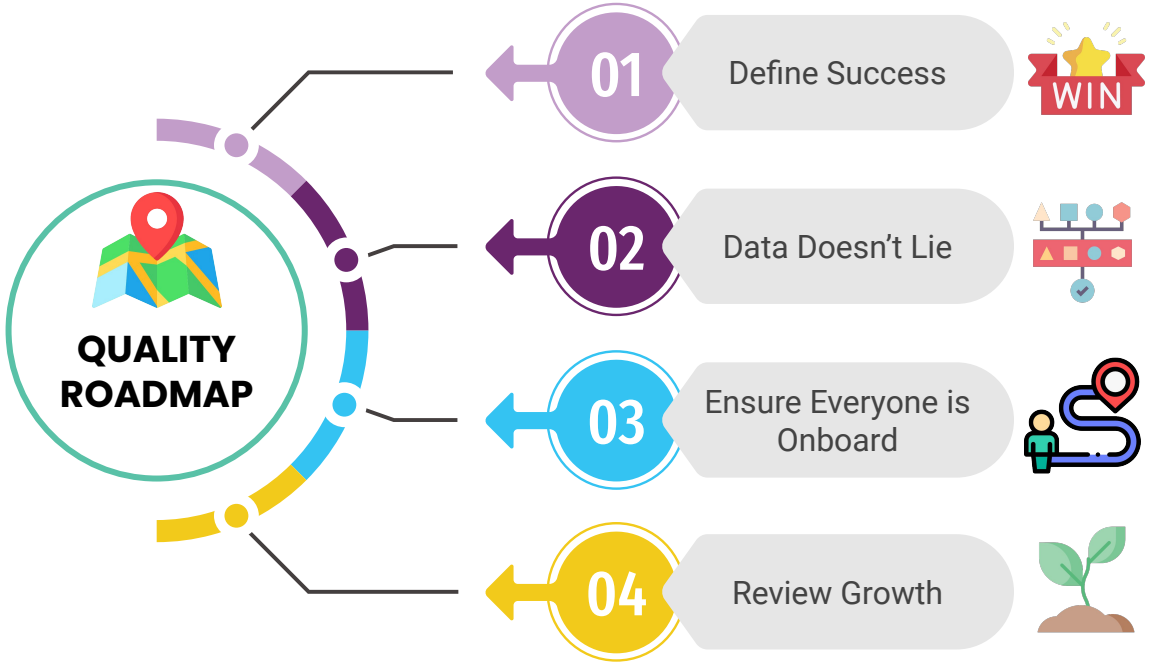
- Check your data and present on issues you may have found with data integrity.
- Ensure you don't make data mistakes in the future (Discuss on how are you going to do this). Trust is important.
- Does this initiative make sense to run anymore given we had bad assumptions?

PRIORITIZATION CHANGES

- Recalculate how to Scale your project and see if it makes sense to continue.
- Don't jump onto a new project without understanding goals and tradeoffs.
- Ensure you have a backlog of other projects to pivot to (ranked off impact)

RECAP

TAKEAWAYS FROM TODAY



Connecting the Dots

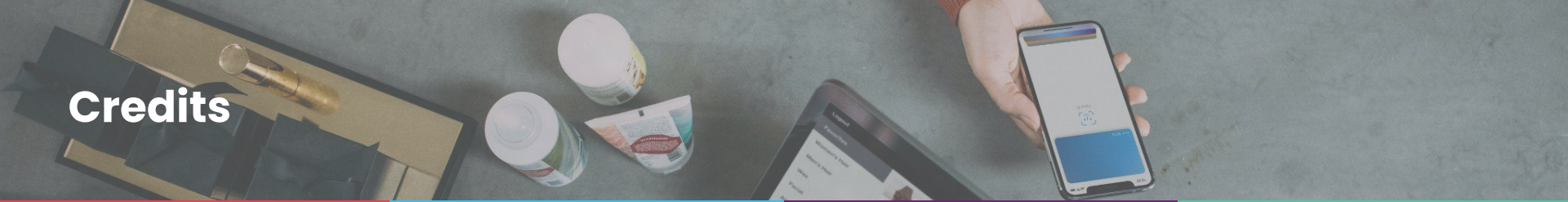
Use a Service Delivery Reviews to Drive Effective Quality Roadmapping

Thank You!

Please stay in touch!

linkedin.com/in/jeffsing

jeff.sing@iterable.com



Credits

Images

- Columbus - Mike Beaumont
- Fog - Katie Moum
- Heaven's Throne Room - Ian Stauffer
- Confusion - Jon Tyson
- Crack me up - Tom Barret
- Speak Up - Designecologist
- Roots - Felix Mittermeier
- iPhone - Tamaz Tuzes-Katai
- Odesza - Dominic Hampton
- Italia - Daniele Colucci
- Chopping Ingredients - Katie Smith
- Map with colorful pins - delfi de la Rua
- Charting Goals - Issac Smith
- Budapest - Josef Keller