QEX: Automated Testing Observability and QA Developer Experience Framework

16th IEEE International Conference on Software Testing, Verification and Validation (ICST) 2023

***Authors: Luohua Huang**, Joseph Chu, Keshia Yap, Hock Yao Chua*

Contents

- Background & Motivation
- Approach
- Metrics, Framework & Implementation
- Evaluation and Analysis
- Challenges Faced
- Conclusion & Future work
- Q&A

Background & Motivation

Continuous-Integration Continuous Delivery

Frequent release of functional product code is heavily reliant on smooth and high quality automated regression testing.

Observability

Acquire reliable metrics on test automation development, execution and its impact on the speed and quality of software development

Impact of QAs

Maturity of testing framework directly impacts production speed and efficiency

Individual QA Experience

Acquire a quantifiable visualisation on contributions, affecting the “cognition” and “conation” of each test automation contributor.

Flexibility

Able to accommodate the best test tools for each product over time in the fast-changing technology field

Background & Motivation

QEX

Monitoring Framework

Comprehensive, reliable and timely feedback on testing activity

Integrate common testing data sources for high observability

Scalable and suitable for large-scale enterprises

Approach & Considerations

Main Objectives

- Quantify usage, stability and development of automation test frameworks
- Understand performance of QA teams
- Provide actionable insights to improve testing infrastructure

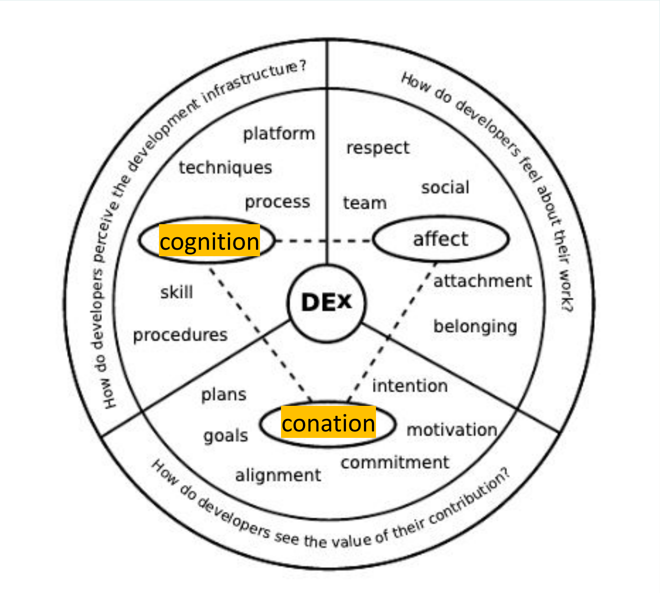
Approach & Considerations

Test Development & Execution Statistics

- Data derived directly from test development and execution
  - Number of automation tests developed & executed
  - Execution time trends
  - Pass rate trends
  - Flakiness

- Accurate and measurable insights into quality and reliability of testware

- Contribute to the *cognition* and *conation* aspects of QA's Developer Experience[1]


How do developers perceive the development infrastructure?
How do developers feel about their work?
How do developers see the value of their contribution?
platform
techniques
process
cognition
skill
procedures
respect
social
team
affect
attachment
belonging
DEx
plans
goals
conation
alignment
intention
motivation
commitment

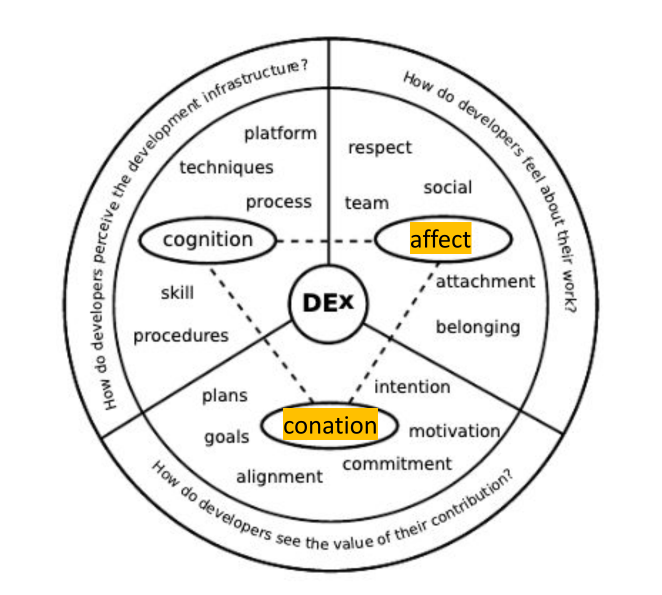
Developer experience: Conceptual framework[1]

[1] Fabian Fagerholm; Jürgen Münch (June 2012). Developer Experience: Concept and Definition. Retrieved from DOI:10.1109/ICSSP.2012.6225984 on 17 April 2023.

Approach & Considerations

Downstream Statistics

- Measures statistics like
  - bugs caught by automation tests
  - Number of tickets signed off by QA
- Quantifies impact of automation tests and provides feedback to QAs
- Contribute to *affect* and *conation* aspects of QA's Developer Experience

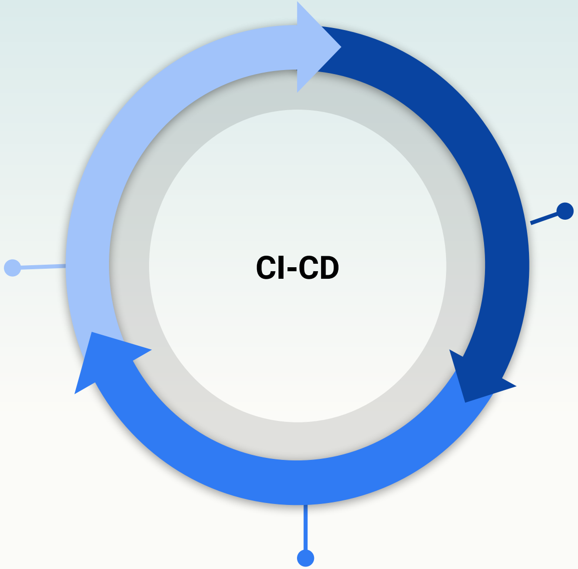

How do developers perceive the development infrastructure?
How do developers feel about their work?
How do developers see the value of their contribution?
platform
techniques
process
respect
social
team
cognition
affect
DEx
skill
procedures
attachment
belonging
plans
goals
alignment
conation
intention
motivation
commitment

Developer experience: Conceptual framework[1]

[1] Fabian Fagerholm; Jürgen Münch (June 2012). Developer Experience: Concept and Definition. Retrieved from DOI:10.1109/ICSSP.2012.6225984 on 17 April 2023.

Approach & Considerations

Examples of desired metrics



Downstream

- Number of tickets signed off
- Number of bugs found
- “Freshness” of automated tests
- Assessing coverage across different products/scopes

Test Development

- Coverage of automated tests or which area requires coverage
- Frequency of code change for test
- Contribution
- Speed of development

Test Execution

- Pass/Fail/Skip rates of cases
- Number of times executed or usage of automated tests
- Flakiness/Stability
- How active the automated tests are

Metrics

QEX metrics

- Active score
  - Automated test cases executed vs total automated test cases available
  - Helps QAs assess the usage of a testing repository
  - Encourages the maintenance of an efficient test code repository with minimal dead code
- Pass rate
  - Percentage of test cases executed that passed
  - Helps to measure test flakiness
- Stability score
  - Compound metric based on active score and pass rate
  - Automated tests with high stability score is effective for flagging potential bugs in the product code

Framework

Breaking it down

- Commonly used components in any testware
  - Code repository, e.g, Git
  - Testing platform, e.g, Jenkins
  - Task management platform, e.g, Jira
- Use an event-driven architecture
- Obtain data from the source
- Use open interfaces to ensuring versatility

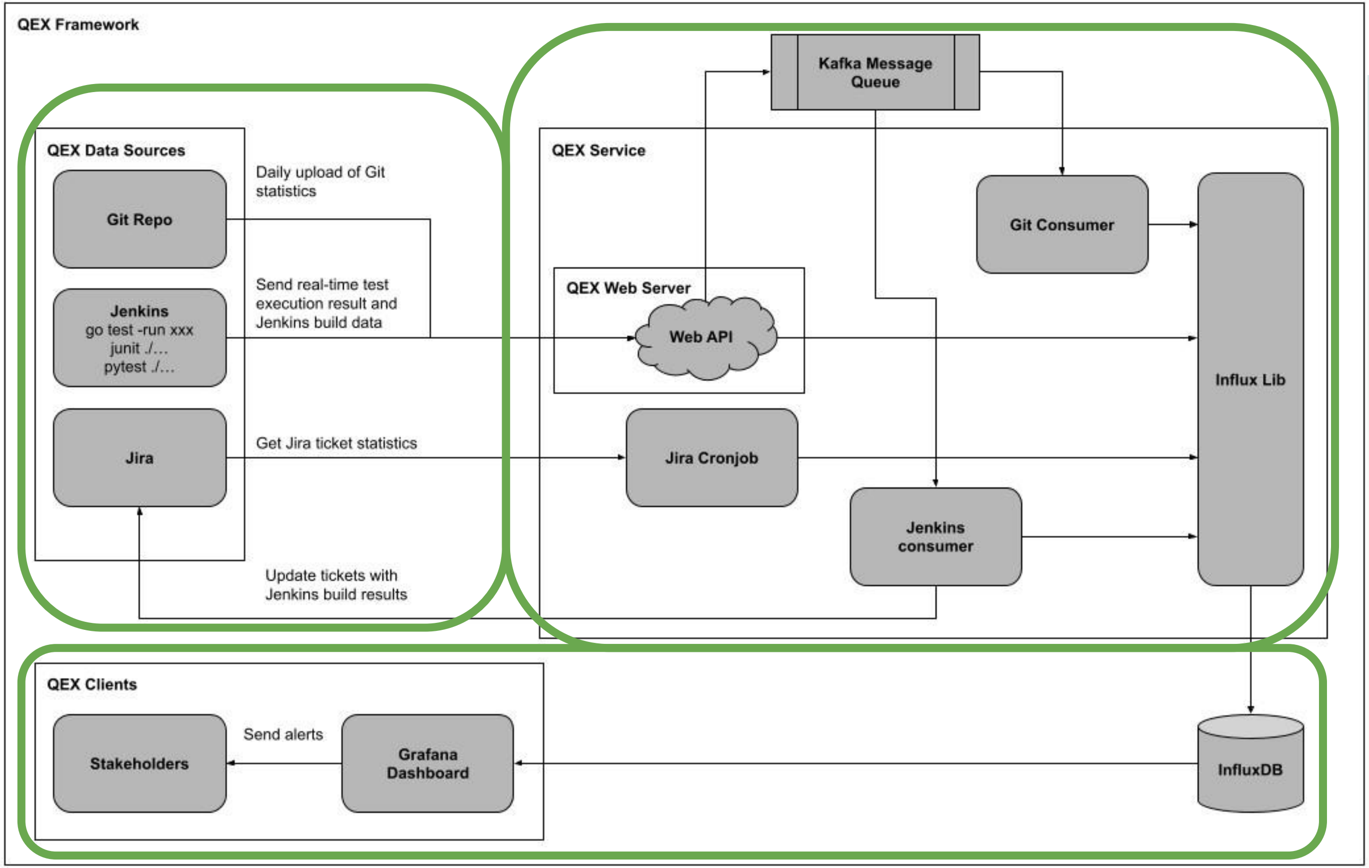
Implementation

Tools and interfaces

- Selected based on current testware used and open-source tools
- RESTful API to transmit test data
- Tools and libraries used
  - Gitlab
  - Jira
  - Jenkins
  - Golang test library
  - InfluxDB
  - Kafka

Implementation

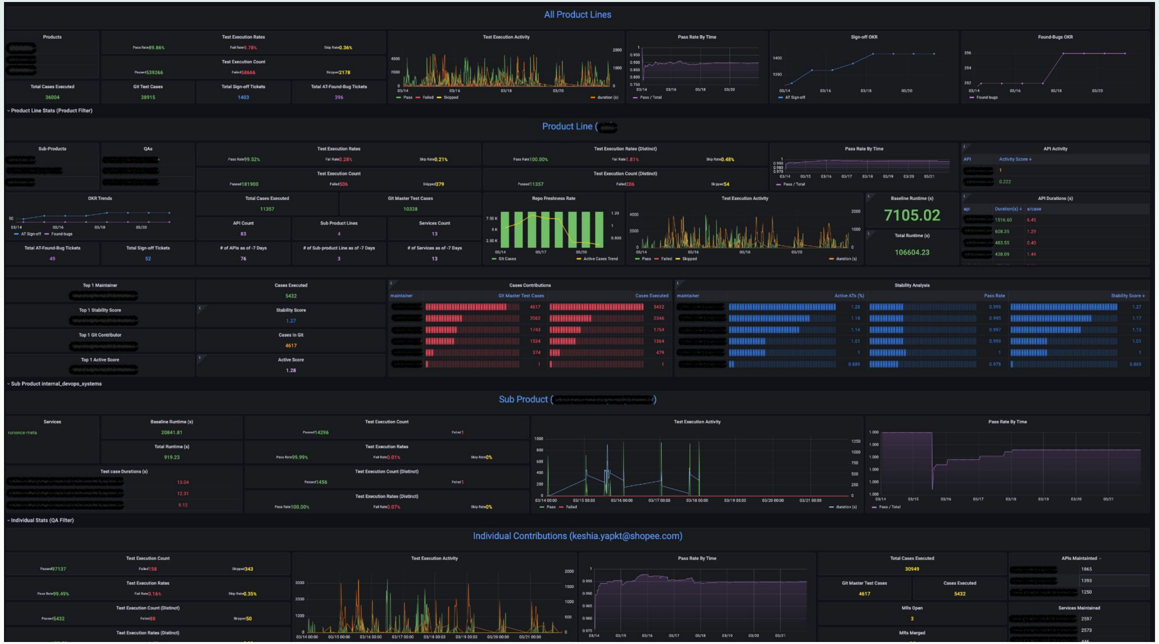
Collection and Processing of data



Implementation

Visualisation and Monitoring

- All data and calculated metrics are displayed in real-time through Grafana Dashboard
  - All Product Lines
  - Individual Product Line
  - Individual Sub-Product
  - Individual Contributor



All Product Lines

Products: [redacted]

Test Execution Rates: Pass Rate 89.86% | Fail Rate 9.78% | Skip Rate 0.36%

Test Execution Count: Passed 539266 | Failed 58666 | Skipped 2178

Total Cases Executed: 36004 | Git Test Cases: 38915 | Total Sign-off Tickets: 1403 | Total AT-Found-Bug Tickets: 396

Test Execution Activity | Pass Rate By Time | Sign-off OKR | Found-Bugs OKR

Product Line Stats (Product Filter)

Product Line ([redacted]

Sub-Products: [redacted] | QAs: [redacted]

Test Execution Rates: Pass Rate 99.52% | Fail Rate 0.28% | Skip Rate 0.21%

Test Execution Count: Passed 181900 | Failed 506 | Skipped 379

Test Execution Rates (Distinct): Pass Rate 100.00% | Fail Rate 1.81% | Skip Rate 0.48%

Test Execution Count (Distinct): Passed 11357 | Failed 206 | Skipped 54

Pass Rate By Time

API Activity

API	Activity Score
[redacted]	1
[redacted]	0.222

OKR Trends

Total Cases Executed: 11357 | Git Master Test Cases: 10328

API Count: 83 | Sub Product Lines: 4 | Services Count: 13

Total AT-Found-Bug Tickets: 49 | Total Sign-off Tickets: 52 | # of APIs as of -7 Days: 76 | # of Sub-product Line as of -7 Days: 3 | # of Services as of -7 Days: 13

Repo Freshness Rate | Test Execution Activity

Baseline Runtime (s): 7105.02

Total Runtime (s): 106604.23

API Durations (s)

api	Duration(s)	s/case
[redacted]	1516.60	6.45
[redacted]	608.35	1.29
[redacted]	483.55	0.40
[redacted]	438.09	1.44

Top 1 Maintainer: [redacted] | Cases Executed: 5432

Top 1 Stability Score: [redacted] | Stability Score: 1.27

Top 1 Git Contributor: [redacted] | Cases In Git: 4617

Top 1 Active Score: [redacted] | Active Score: 1.28

Cases Contributions

maintainer	Git Master Test Cases	Cases Executed
[redacted]	4617	5432
[redacted]	2062	2346
[redacted]	1743	1754
[redacted]	1534	1364
[redacted]	374	479
[redacted]	1	1

Stability Analysis

maintainer	Active ATs (%)	Pass Rate	Stability Score
[redacted]	1.28	0.995	1.27
[redacted]	1.18	0.995	1.17
[redacted]	1.14	0.997	1.13
[redacted]	1.01	0.999	1.01
[redacted]	1	1	1
[redacted]	0.889	0.978	0.869

Sub Product internal_devops_systems

Sub Product ([redacted])

Services: runonce-meta

Baseline Runtime (s): 20841.81

Total Runtime (s): 919.23

Test Execution Count: Passed 14296 | Failed 1

Test Execution Rates: Pass Rate 99.99% | Fail Rate 0.01% | Skip Rate 0%

Test Execution Count (Distinct): Passed 1456 | Failed 1

Test Execution Rates (Distinct): Pass Rate 100.00% | Fail Rate 0.07% | Skip Rate 0%

Test case Durations (s)

[redacted]	13.04
[redacted]	12.31
[redacted]	9.12

Test Execution Activity | Pass Rate By Time

Individual Stats (QA Filter)

Individual Contributions (keshia.yapkt@shopee.com)

Test Execution Count: Passed 97137 | Failed 158 | Skipped 343

Test Execution Rates: Pass Rate 99.49% | Fail Rate 0.16% | Skip Rate 0.35%

Test Execution Count (Distinct): Passed 5432 | Failed 88 | Skipped 50

Test Execution Rates (Distinct)

Test Execution Activity | Pass Rate By Time

Total Cases Executed: 30949

Git Master Test Cases: 4617 | Cases Executed: 5432

MRs Open: 3

MRs Merged

APIs Maintained

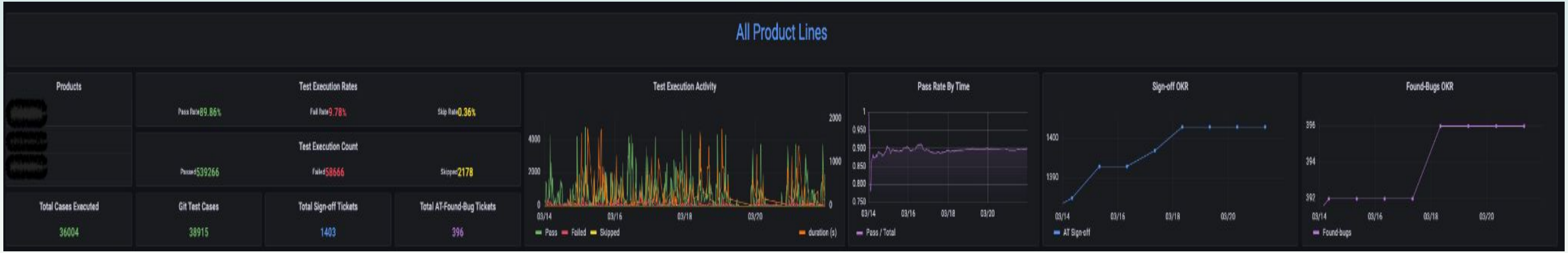
[redacted]	1865
[redacted]	1393
[redacted]	1250

Services Maintained

[redacted]	2597
[redacted]	2573

Implementation

Visualisation and Monitoring - All Product Lines



All Product Lines

Products

[redacted]

Test Execution Rates

Pass Rate 89.86% Fail Rate 9.78% Skip Rate 0.36%

Test Execution Count

Passed 539266 Failed 58666 Skipped 2178

Total Cases Executed 36004

Git Test Cases 38915

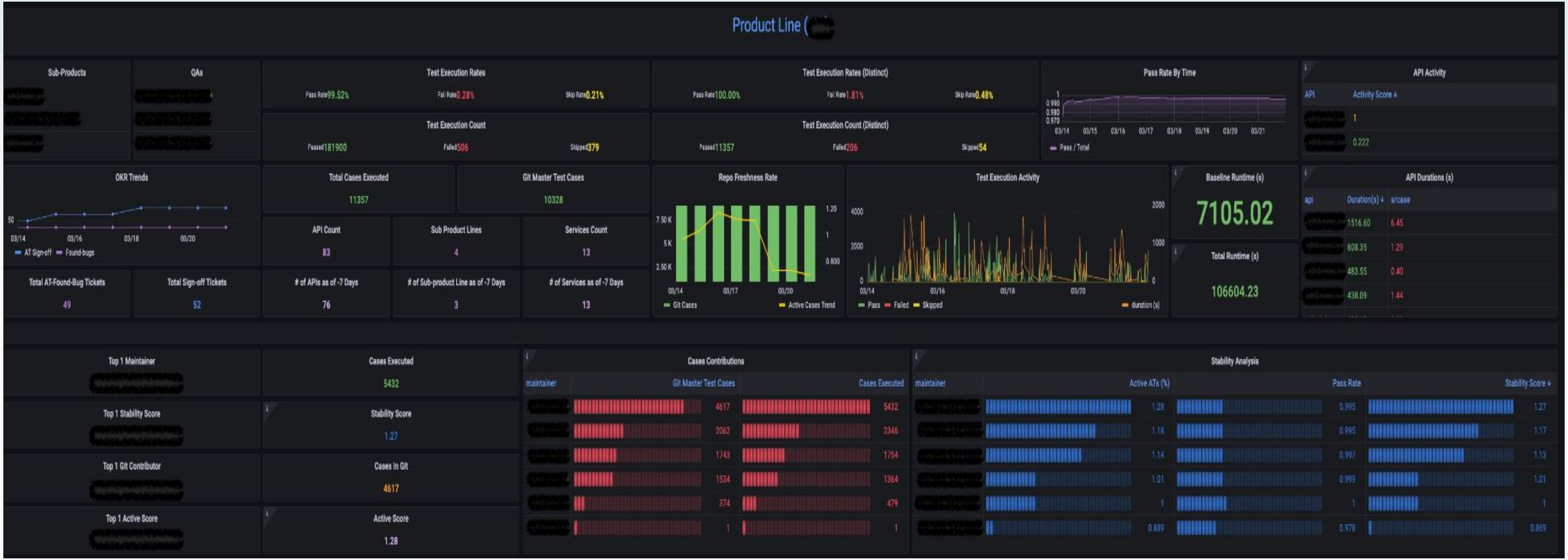
Total Sign-off Tickets 1403

Total AT-Found-Bug Tickets 396

- Overall Metrics
  - Pass / Fail Rate
  - Pass Rate By Time
  - Number of Cases Executed
  - Tickets Opened
  - etc.

Implementation

Visualisation and Monitoring - Individual Product Lines



Product Line ([redacted]

Sub-Products

[redacted]

QAs

[redacted]

Test Execution Rates

Pass Rate 99.52% Fail Rate 0.28% Skip Rate 0.21%

Test Execution Count

Passed 181900 Failed 506 Skipped 379

Test Execution Rates (Distinct)

Pass Rate 100.00% Fail Rate 1.81% Skip Rate 0.48%

Test Execution Count (Distinct)

Passed 11357 Failed 206 Skipped 54

Pass Rate By Time

API Activity

API	Activity Score ↓
[redacted]	1
[redacted]	0.222

OKR Trends

Total Cases Executed 11357

Git Master Test Cases 10328

API Count 83

Sub Product Lines 4

Services Count 13

Total AT-Found-Bug Tickets 49

Total Sign-off Tickets 52

# of APIs as of -7 Days 76

# of Sub-product Line as of -7 Days 3

# of Services as of -7 Days 13

Repo Freshness Rate

Test Execution Activity

Baseline Runtime (s) 7105.02

Total Runtime (s) 106604.23

API Durations (s)

api	Duration(s) ↓	s/case
[redacted]	1516.60	6.45
[redacted]	608.35	1.29
[redacted]	483.55	0.40
[redacted]	438.09	1.44

Top 1 Maintainer [redacted]

Cases Executed 5432

Top 1 Stability Score [redacted]

Stability Score 1.27

Top 1 Git Contributor [redacted]

Cases In Git 4617

Top 1 Active Score [redacted]

Active Score 1.28

Cases Contributions

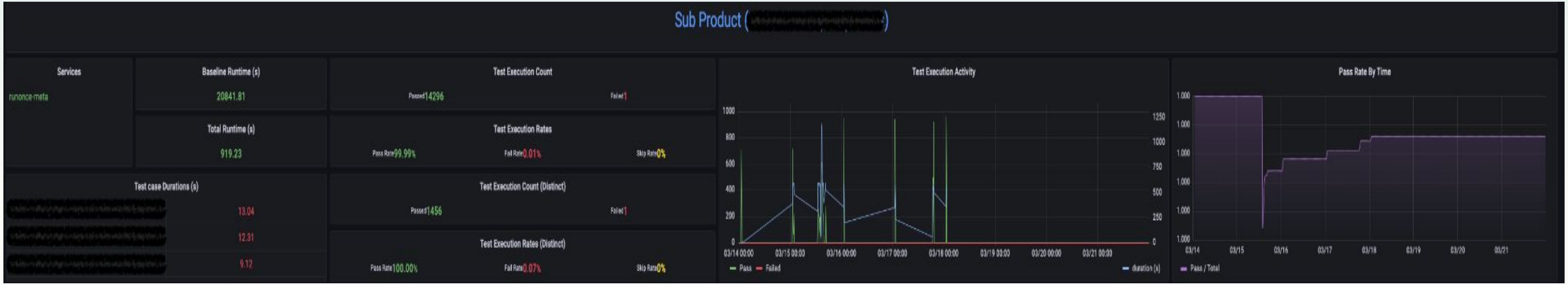
maintainer	Git Master Test Cases	Cases Executed
[redacted]	4617	5432
[redacted]	2062	2346
[redacted]	1743	1754
[redacted]	1534	1364
[redacted]	374	479
[redacted]	1	1

Stability Analysis

maintainer	Active ATs (%)	Pass Rate	Stability Score ↓
[redacted]	1.28	0.995	1.27
[redacted]	1.18	0.995	1.17
[redacted]	1.14	0.997	1.13
[redacted]	1.01	0.999	1.01
[redacted]	1	1	1
[redacted]	0.889	0.978	0.869

Implementation

Visualisation and Monitoring - Individual Sub-Products



Sub Product ([redacted])

Services

runonce-meta

Baseline Runtime (s)

20841.81

Total Runtime (s)

919.23

Test case Durations (s)

[redacted]	13.04
[redacted]	12.31
[redacted]	9.12

Test Execution Count

Passed 14296 Failed 1

Test Execution Rates

Pass Rate 99.99% Fail Rate 0.01% Skip Rate 0%

Test Execution Count (Distinct)

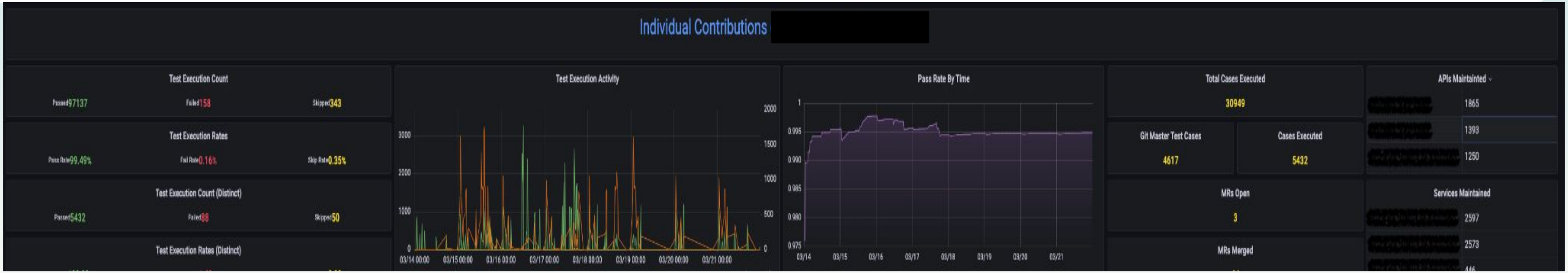
Passed 1456 Failed 1

Test Execution Rates (Distinct)

Pass Rate 100.00% Fail Rate 0.07% Skip Rate 0%

Implementation

Visualisation and Monitoring - Individual Contributor



Individual Contributions [redacted]

Test Execution Count

Passed97137 Failed158 Skipped343

Test Execution Rates

Pass Rate99.49% Fail Rate0.16% Skip Rate0.35%

Test Execution Count (Distinct)

Passed5432 Failed88 Skipped50

Test Execution Rates (Distinct)

Total Cases Executed

30949

Git Master Test Cases

4617

Cases Executed

5432

MRs Open

3

MRs Merged

APIs Maintainted ⌄	
[redacted]	1865
[redacted]	1393
[redacted]	1250

Services Maintained	
[redacted]	2597
[redacted]	2573
[redacted]	446

- Individual Contributor Feedback
  - Stability Score
  - Active Score
  - Contribution of test cases
  - etc.

Evaluation and Analysis

Performance and Usage

- Supports over 100 QA engineers across 10 product lines, running over 15,000 automation tests daily

- QEX downtime is minimal (97% Uptime)
  - Notification system ensures quick recovery

- Real-time data
  - Less than 1 second for each test-run

- Stream processors and cache helps minimize data loss/duplication

Challenges Faced

Selection and Usage of Metrics

- Users should be mindful of how metrics are interpreted and used
- Quantitative measurements should be considered in context
- Be explicit to your team in how metrics are used and how they relate to the team’s OKR

Challenges Faced

Barriers to Adoption - Technical Complexity

- Multi-component framework
  - Made simpler by streamlining deployment process
- Requires a certain level of technical ability to set up and maintain
  - Setup is a one time effort
  - Modularized architecture is easier to maintain

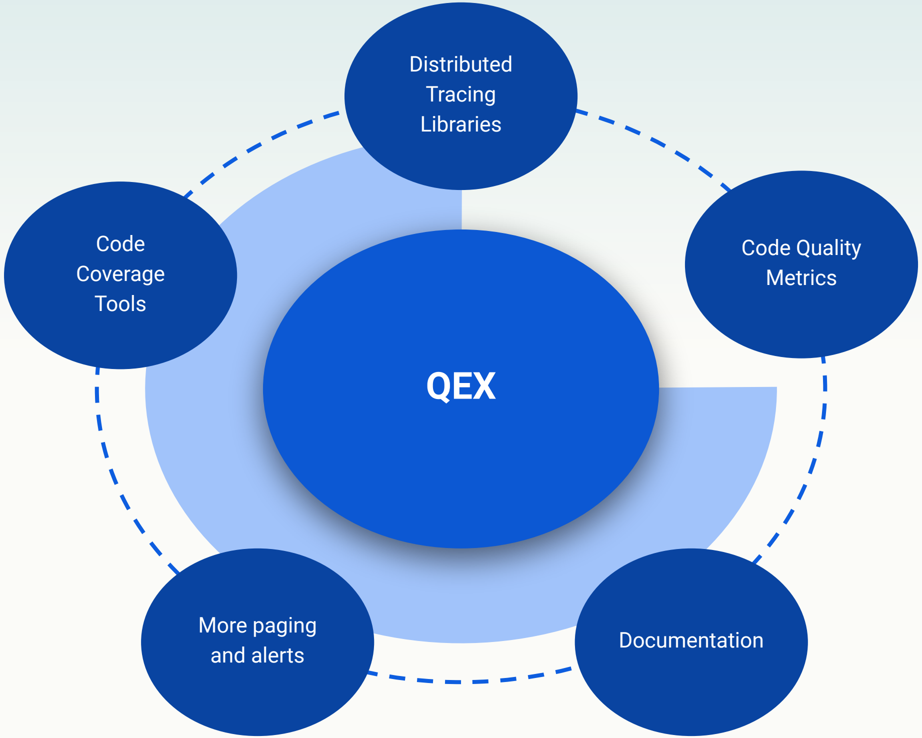
Challenges Faced

Barriers to Adoption - Cost of Customised Tools

- Maintenance effort is not small
- Might require additional workarounds for certain frameworks (e.g. Web-based testing)
- Tradeoff for sensitivity and real-time data

Conclusion & Future work

- QEX's open architecture - Scale easily and integrate with other tools or systems to have better test observability



Thank You!

Q&A