LiveVideo StackCon 音视频技术大会

技术开启新“视”界
Technology Bring New Vision

Guard the Media Quality in Modern Software Development, from Automated Testing to Metric Driven Product Monitoring

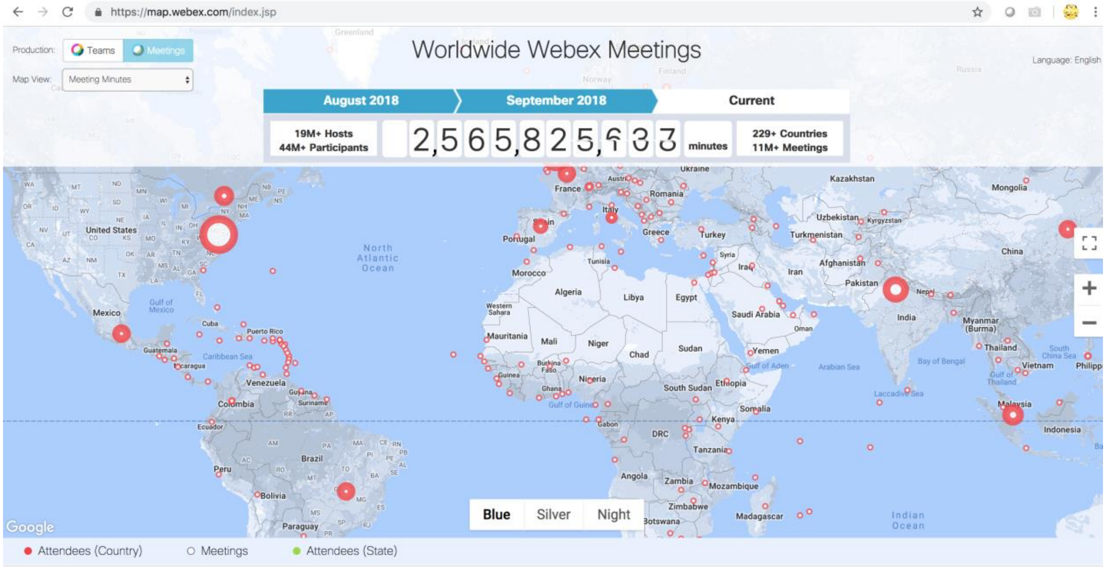
Rui Zhang
Principle Engineer
Cisco Systems

LiveVideoStack
音视频技术社区

CSDN

1 Introduction to Cisco Collaboration Business

Introduction to Cisco Collaboration



https://map.webex.com/index.jsp

Production: Teams Meetings

Map View: Meeting Minutes

Worldwide Webex Meetings

Language: English

August 2018 | September 2018 | Current

19M+ Hosts
44M+ Participants

2,565,825,633 minutes

229+ Countries
11M+ Meetings

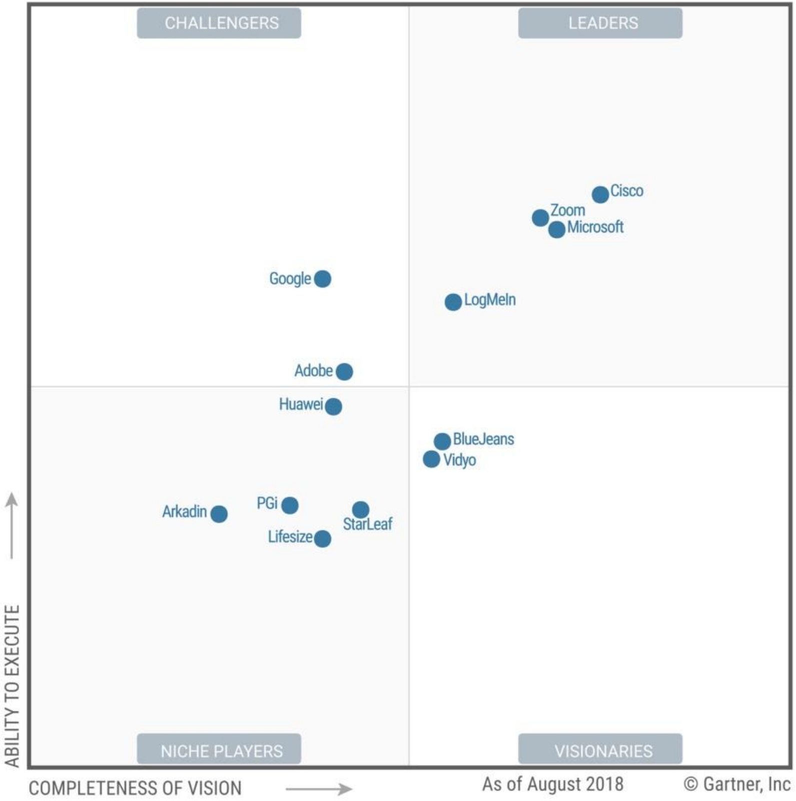
Blue Silver Night

● Attendees (Country) ○ Meetings ● Attendees (State)

- 95% of the Fortune 500 as customers
- 70 billion+ minutes of great ideas happen every year
- 113,000 people coordinates every month

Introduction to Cisco Collaboration

Figure 1. Magic Quadrant for Meeting Solutions



Source: Gartner (September 2018)

1. Quick Introduction about Cisco Collaboration Business
2. Media Quality Control in Agile Software Development
3. Metrics, Events/Statistics, and Signals: All for Quality of Experience
4. Summary and Looking Forward

2 Media Quality Control in Agile Software Development

Media Quality Control in Agile Software Development

Agile Development and Cloud Delivery

- Business fast-changing environment
- Rapid software development
- Continuous Integration and Continuous Delivery
- Iteration to evolve

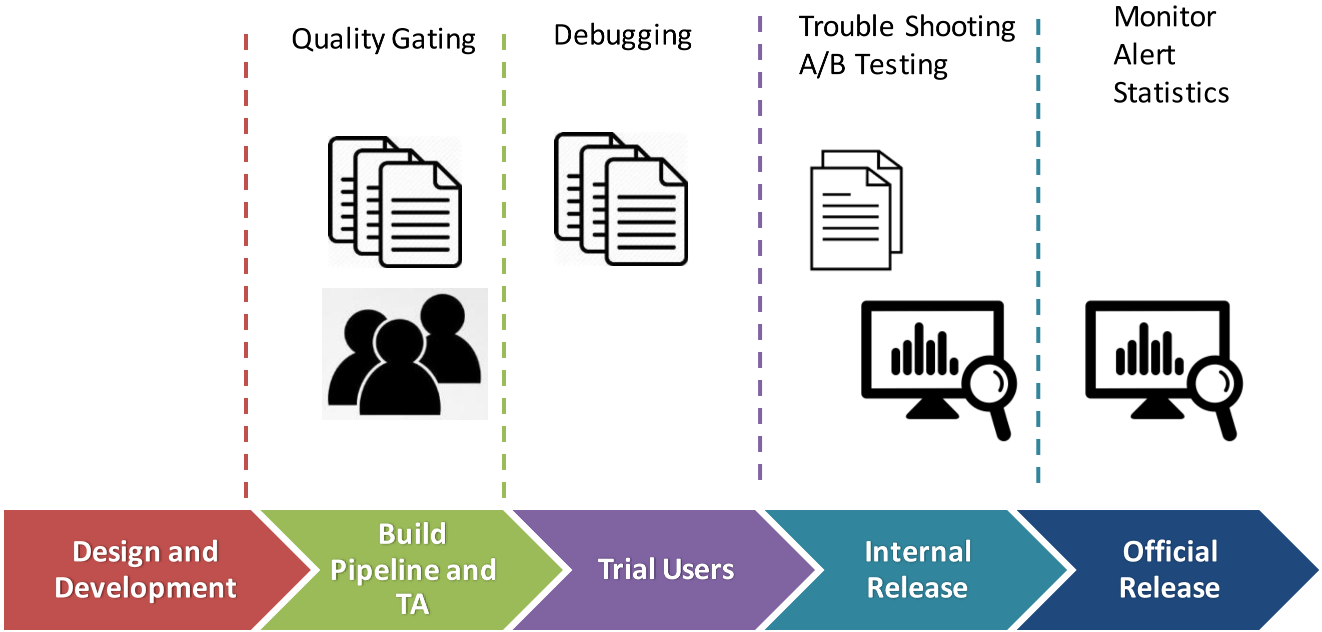
Media Quality Control in Conferencing

- Quality is subjective
- Media is end-to-end. Many components are impacting the quality
- Many factors impact the quality, from device to network

Media Quality Control in Conferencing

Process	Goals	Technology
Right Time	Right Thing	Right Way

Media Quality Control in Conferencing



Quality Gating

Debugging

Trouble Shooting
A/B Testing

Monitor
Alert
Statistics

Design and Development

Build Pipeline and TA

Trial Users

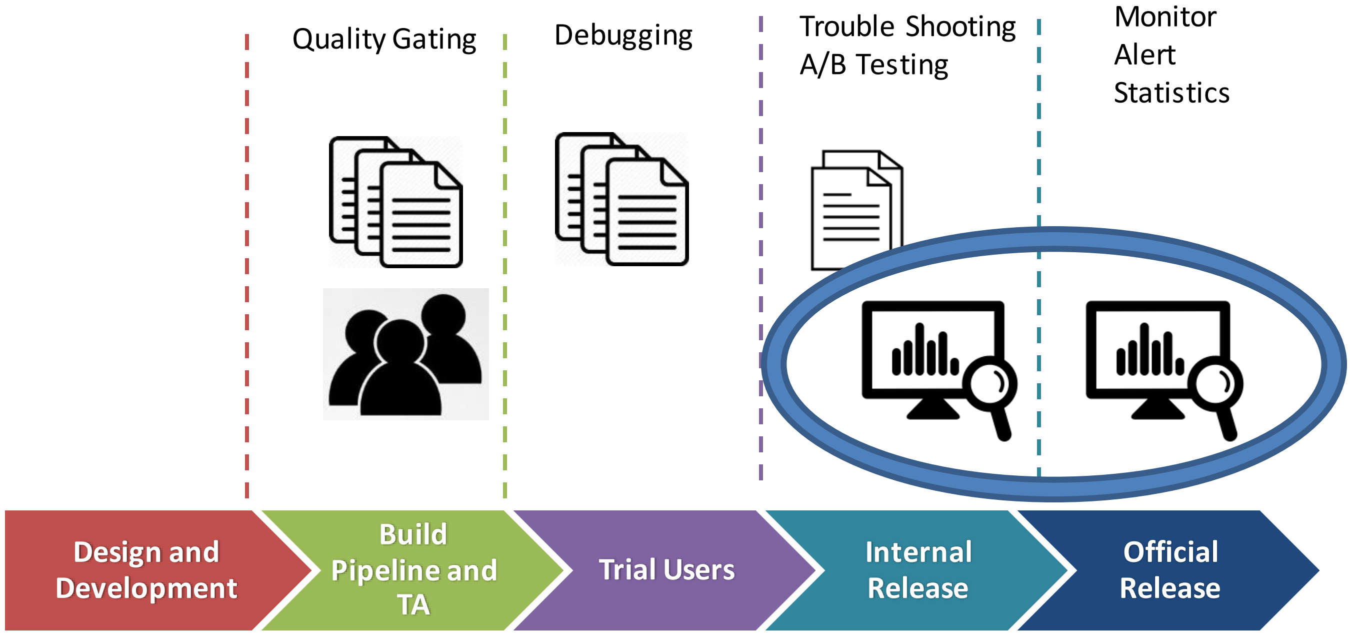
Internal Release

Official Release

3

Metrics, Events/Statistics, and Signals: All for Quality of Experience

Media Quality Control in Conferencing



Quality Gating

Debugging

Trouble Shooting
A/B Testing

Monitor
Alert
Statistics

Design and Development

Build Pipeline and TA

Trial Users

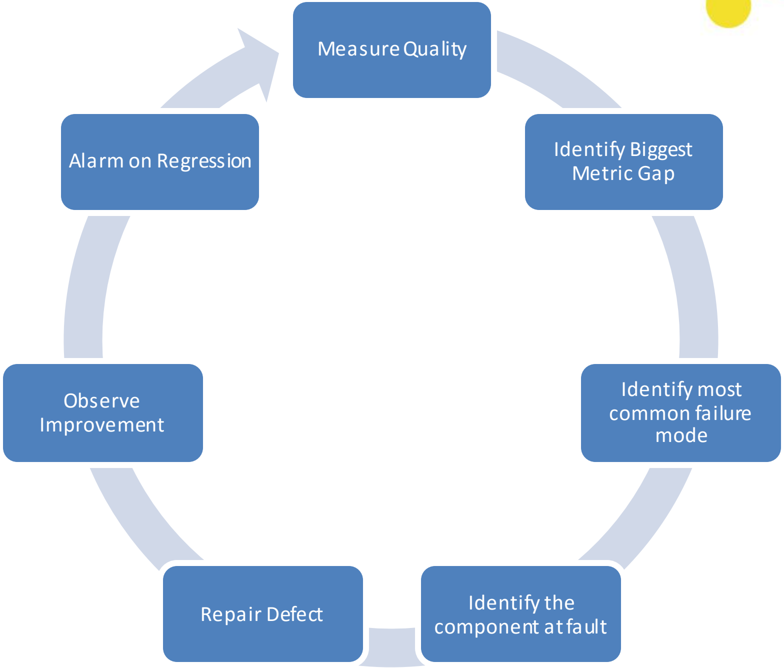
Internal Release

Official Release

Metrics:

- Monitoring
- Alert
- Statistics
- A/B Testing
- All about **Data**, from the overall big picture to corner cases.

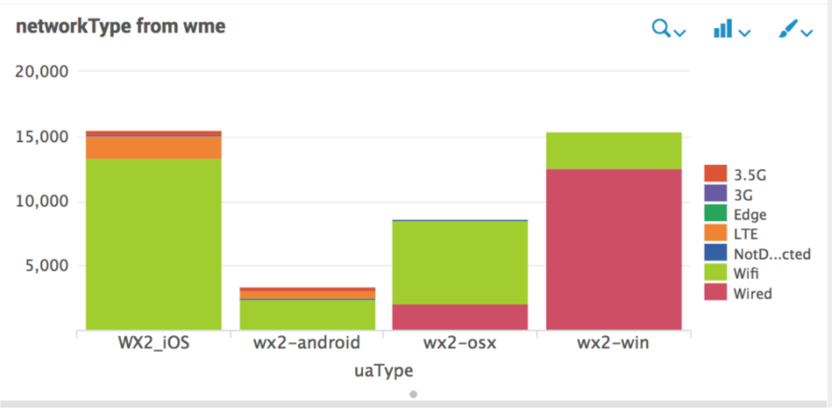
- Raw event
- Statistics
  - Min/max/average
  - Histogram
- Dashboard with dimensioning and sub-casing


Measure Quality
Identify Biggest Metric Gap
Identify most common failure mode
Identify the component at fault
Repair Defect
Observe Improvement
Alarm on Regression

Metrics

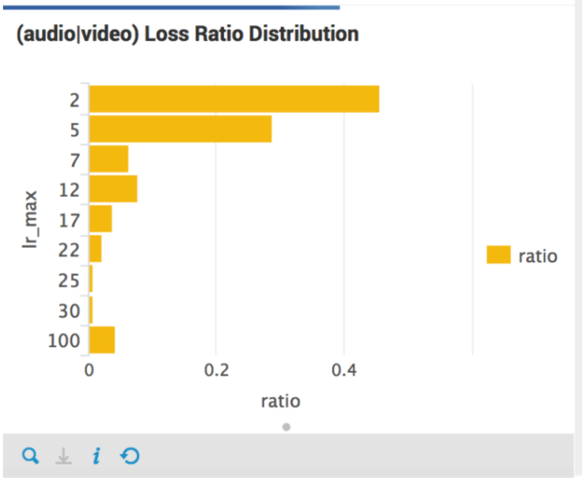
- Understanding the user environments and working style
  - OS distribution
  - A/V device types
  - Network distribution …
- To assist design
  - Network statistics
  - Screen size distribution …
- To monitor/alert
  - Connection failure
  - Resolution/frame rate
  - Decoding out of sync duration
  - Capture/playback error…
  - Acoustic echo cancelation failure…
- To A/B test
  - Media quality
  - Connection delay …

Metrics: Understand User Environment

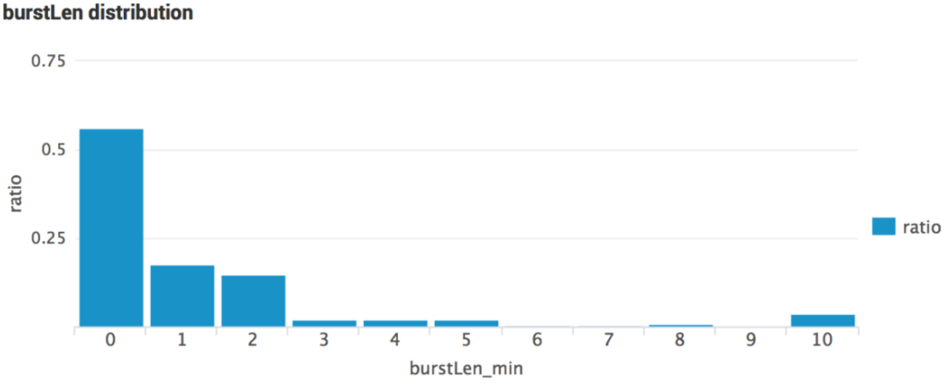


OS/Network Type Distribution

Loss Rate Distribution

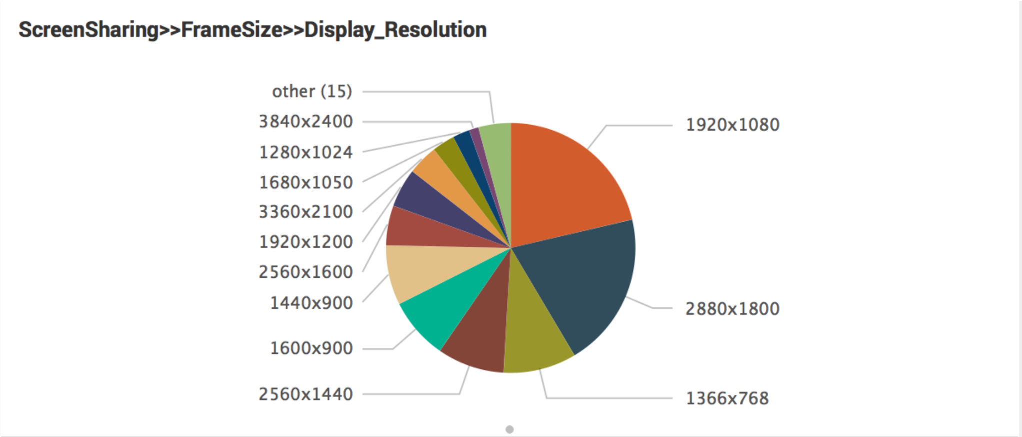


Metrics: To Assist Design

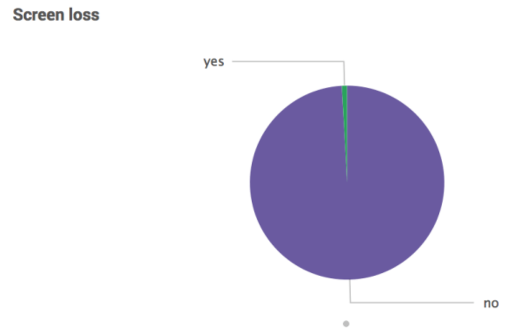


Burst length of packet loss => how we choose FEC code

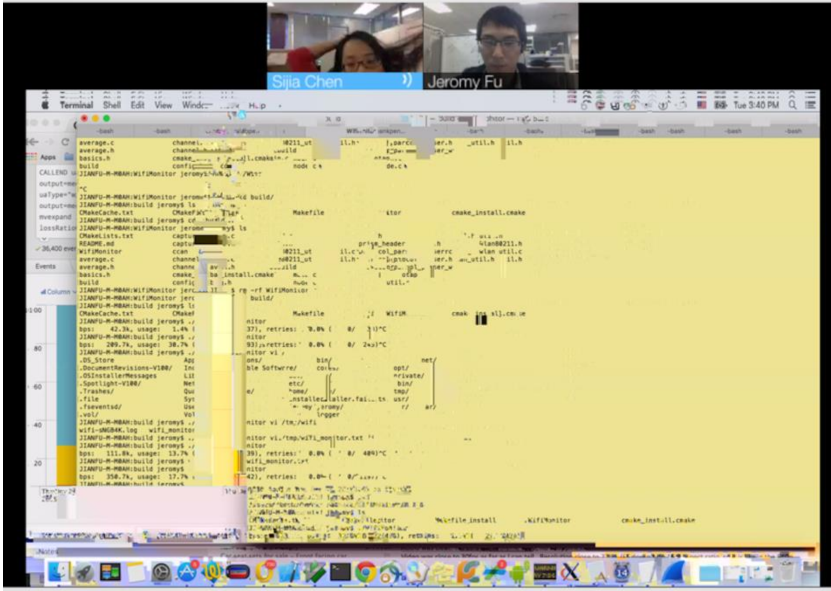
Screen size=> how we design our sharing parameters



Metrics: Monitoring/Alert/Debugging

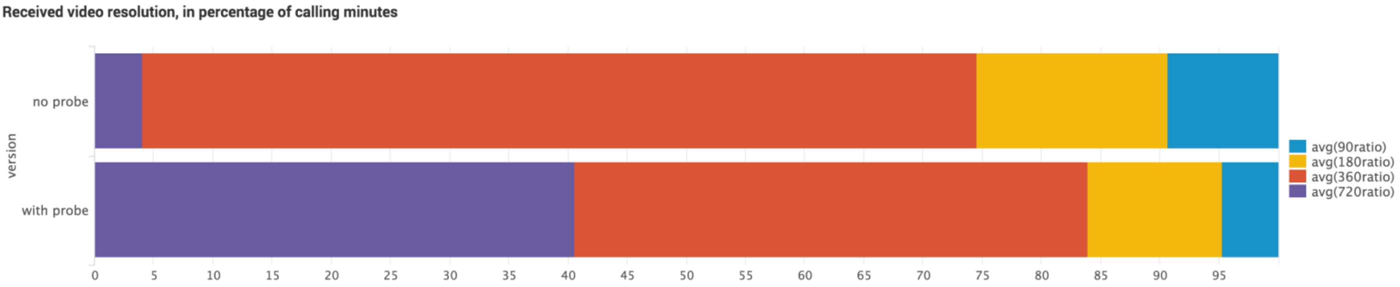


Screen loss



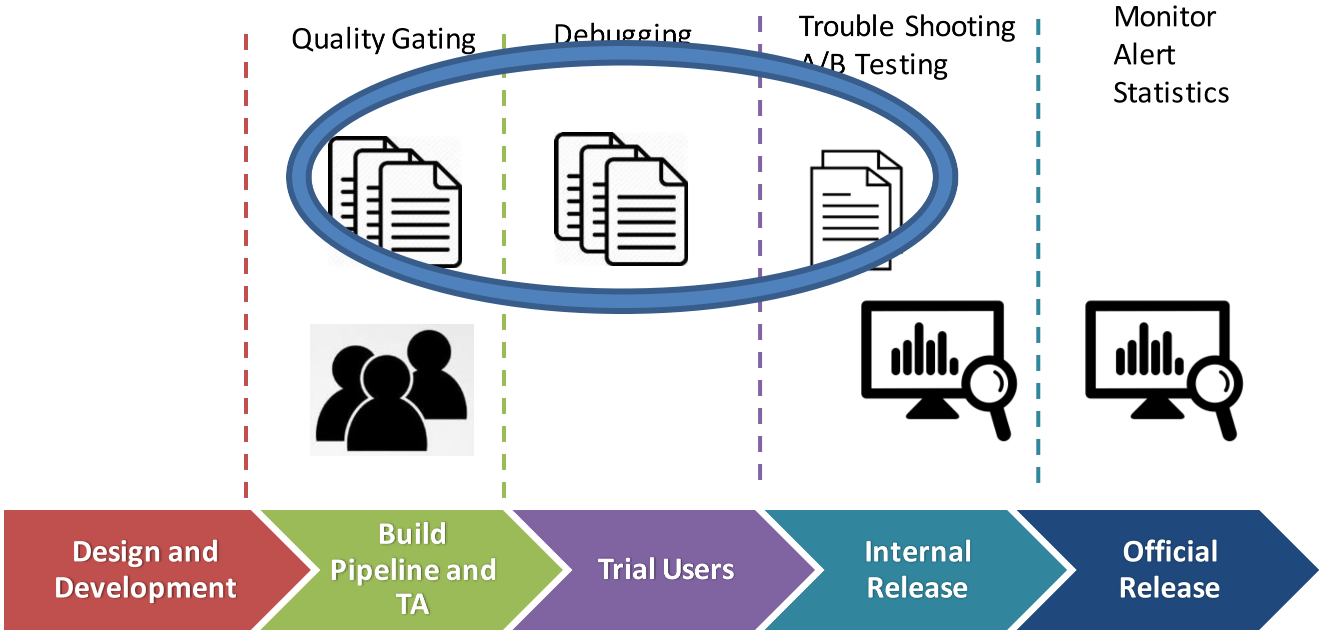
```
average.c           channel.
average.h           channel
basics.h            cmake_
build               confi
JIANFU-M-MBAH:WifiMonitor jeromy$
^C
JIANFU-M-MBAH:WifiMonitor jerom                    build/
JIANFU-M-MBAH:build jeromy$ ls
CMakeCache.txt      CMakeFi                 Makefile          itor          cmake_install.cmake
JIANFU-M-MBAH:build jeromy$ cd ..
JIANFU-M-MBAH:WifiMonitor jero       y$ ls
CMakeLists.txt      captu
README.md           captu                                 prism_header
WifiMonitor         ccan                    80211_ut      col_par         wlan util.c
average.c           channel                 m80211_ut                  er.h   util.h   il.h
average.h           channe                  build
basics.h            cmake_      ba_install.cmake
build               confi                                 util.
JIANFU-M-MBAH:WifiMonitor jer         $ rm -rf WifiMonitor
JIANFU-M-MBAH:WifiMonitor jer                  build/
JIANFU-M-MBAH:build jeromy$ ls
CMakeCache.txt      CMake                   Makefile          WifiM         cmak
JIANFU-M-MBAH:build jeromy$ ./            nitor
bps:    42.3k, usage:   1.4% (            37), retries:   0.0% (    0/   3)^C
JIANFU-M-MBAH:build jeromy$ ./            nitor
bps:   209.7k, usage:  30.7% (            93), retries:   0.0% (    0/  2  )^C
JIANFU-M-MBAH:build jeromy$ ./            nitor vi
.DS_Store           App                   ons/          bin/                net/
.DocumentRevisions-V100/  Inc             ble Softwrre/                     opt/
.OSInstallerMessages      Lit                                               private/
.Spotlight-V100/    Net                              etc/                   bin/
.Trashes/           Qui                   e/         home/                  tmp/
.file               Sys                              installer.failurerequests  usr/
.fseventsd/         Use                              jeromy/                r/
.vol/               Vol                              logger
JIANFU-M-MBAH:build jeromy$ ./            nitor vi /tmp/wifi
wifi-sNGB4K.log     wifi_monitor
JIANFU-M-MBAH:build jeromy$ ..            nitor vi /tmp/wifi_monitor.txt
JIANFU-M-MBAH:build jeromy$ ./            nitor
bps:   111.8k, usage:  13.7% (            39), retries:   0.0% (    0/  409)^C
JIANFU-M-MBAH:build jeromy$               wifi_monitor.txt
JIANFU-M-MBAH:build jeromy$               nitor
bps:   350.7k, usage:  17.7%              42), retries:   0.0% (    0/
```

Metrics: A/B Testing


Received video resolution, in percentage of calling minutes

Video resolution distribution with probing on/off

Media Quality Control in Conferencing



Quality Gating

Debugging

Trouble Shooting
A/B Testing

Monitor
Alert
Statistics

Design and Development

Build Pipeline and TA

Trial Users

Internal Release

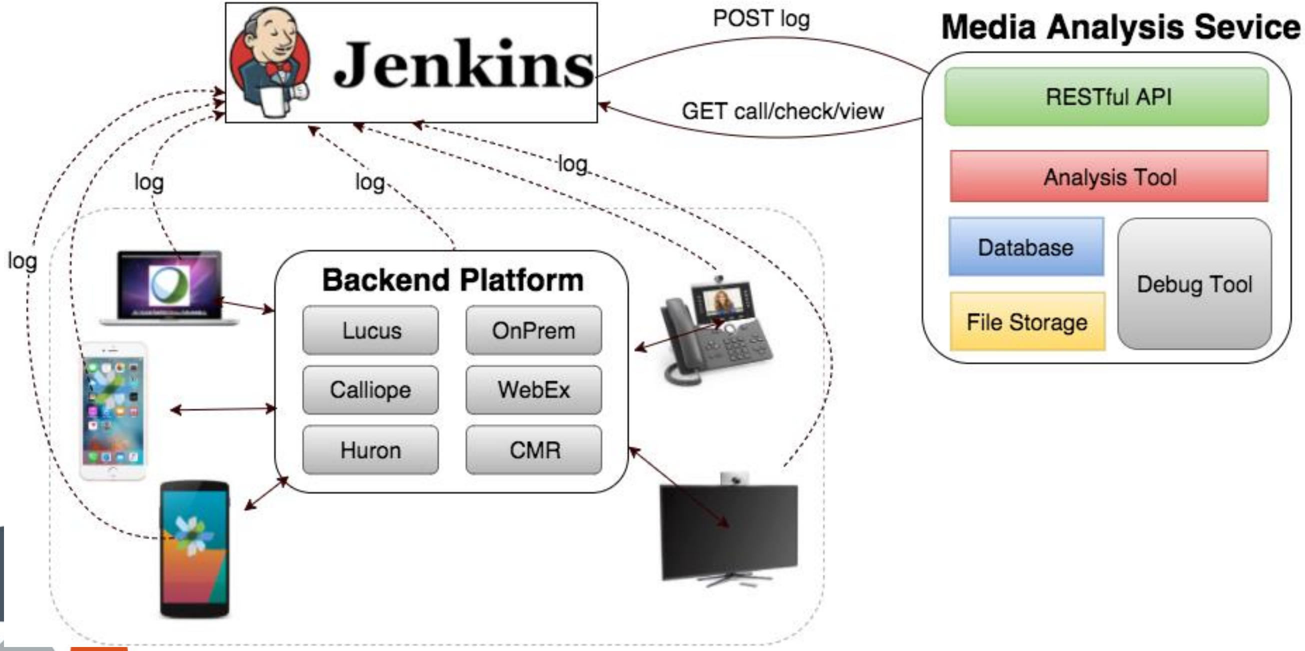
Official Release

Events/Statistics

- Data Collection:
  - Raw event emission from various components (client and server)
  - Logs uploaded from various components
- Data Processing:
  - Correlate on the events/log for the call flow and media flow
  - Monitor the success/failure
  - Monitor the quality
- Identify the troublesome area
  - Call failure?
  - Sharing initial delay?

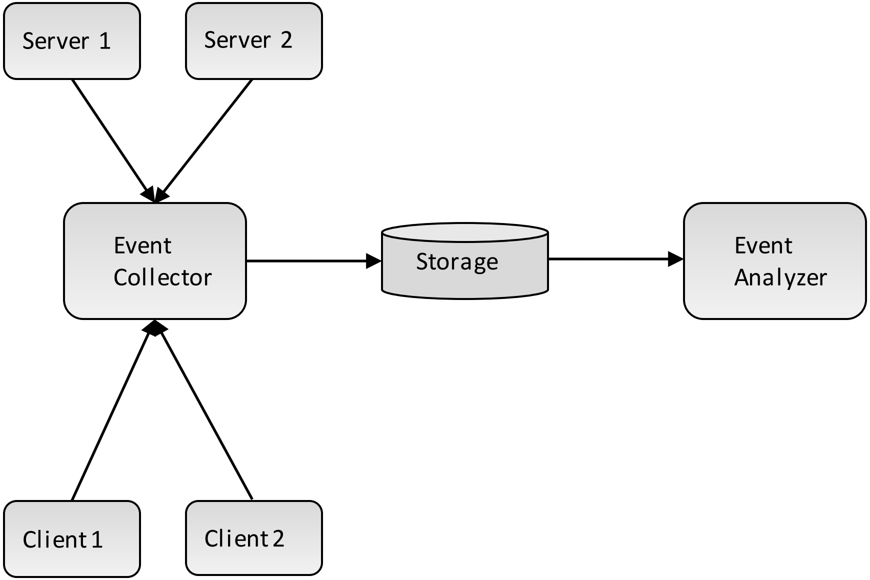
Events/Statistics

Integration with build pipeline



Events/Statistics

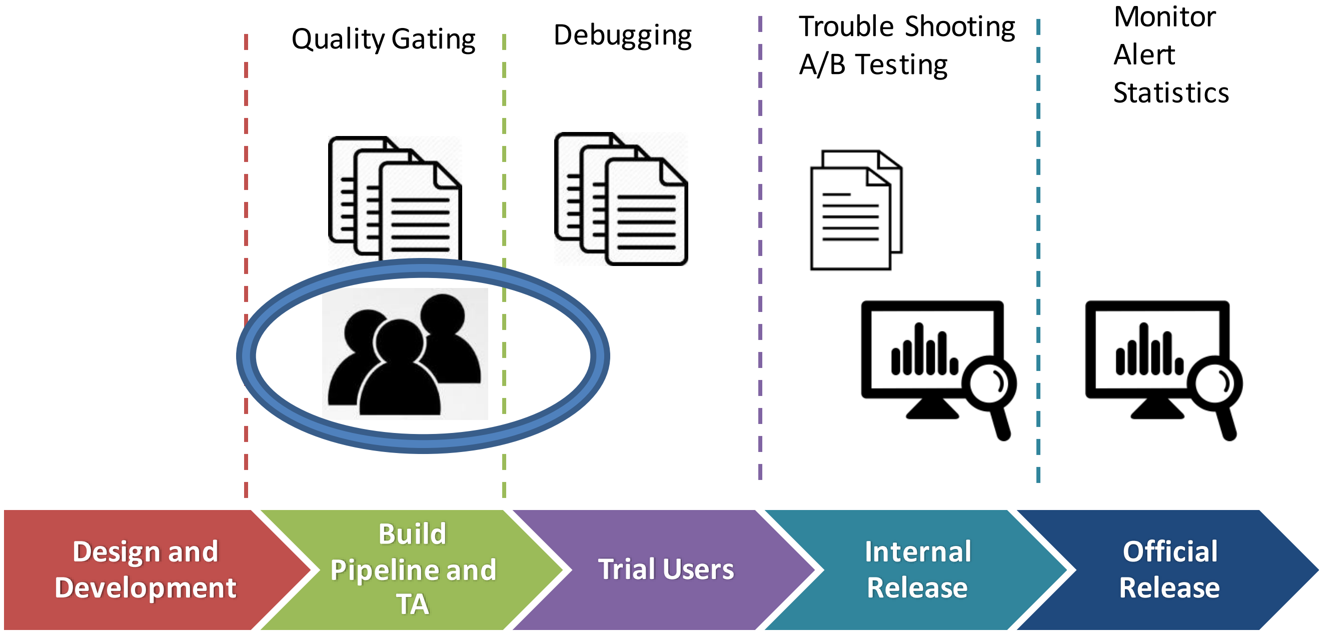
On Production



Events/Statistics

- No video received
  - SDP negotiation?
  - Connectivity set up?
  - Camera started successfully?
  - Encoder worked?
- Video received lower resolution than expected
  - SDP parameters right?
  - Receiver asking right resolution?
  - Server sending right request?
  - Sender sending right resolution?
  - Network good or bad?

Media Quality Control in Conferencing



Quality Gating

Debugging

Trouble Shooting
A/B Testing

Monitor
Alert
Statistics

Design and Development

Build Pipeline and TA

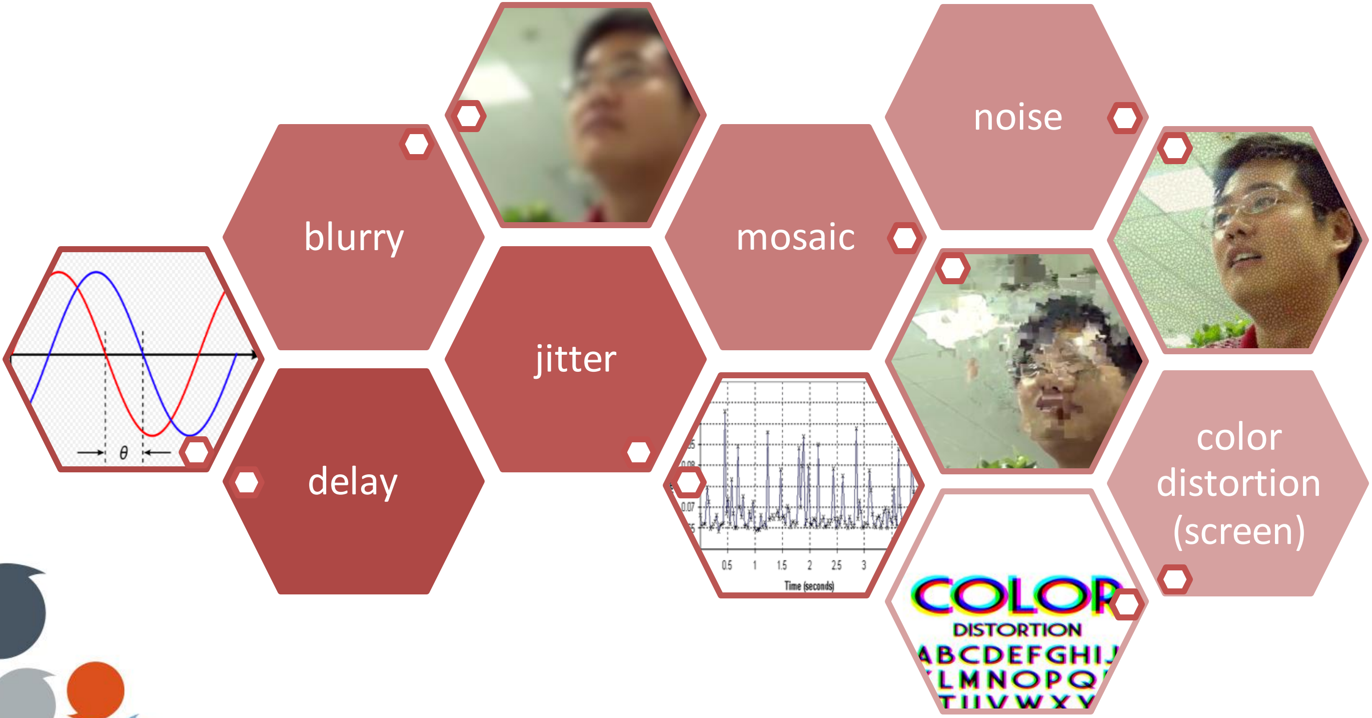
Trial Users

Internal Release

Official Release

Signals

Video deteriorations in multiple dimensions



noise

blurry

mosaic

jitter

delay

color distortion (screen)

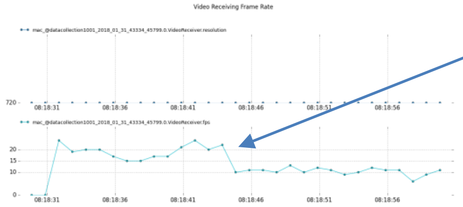
Signals

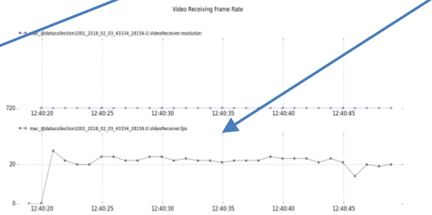
- Build up objective video quality model: one score measurement/assessment
- Covering multi-dimensional video deterioration
- Non-referencing scoring in pixel domain
  - Modeling end-user experience
  - Suitable for black-box comparison
- Referencing scoring, with assistance on log info, to add further measurement
  - Motion smoothness
  - Network impairment
- Continuous calibration with log analysis as well as subjective feedback

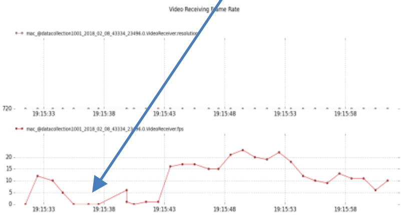
Signals

Video Smoothness

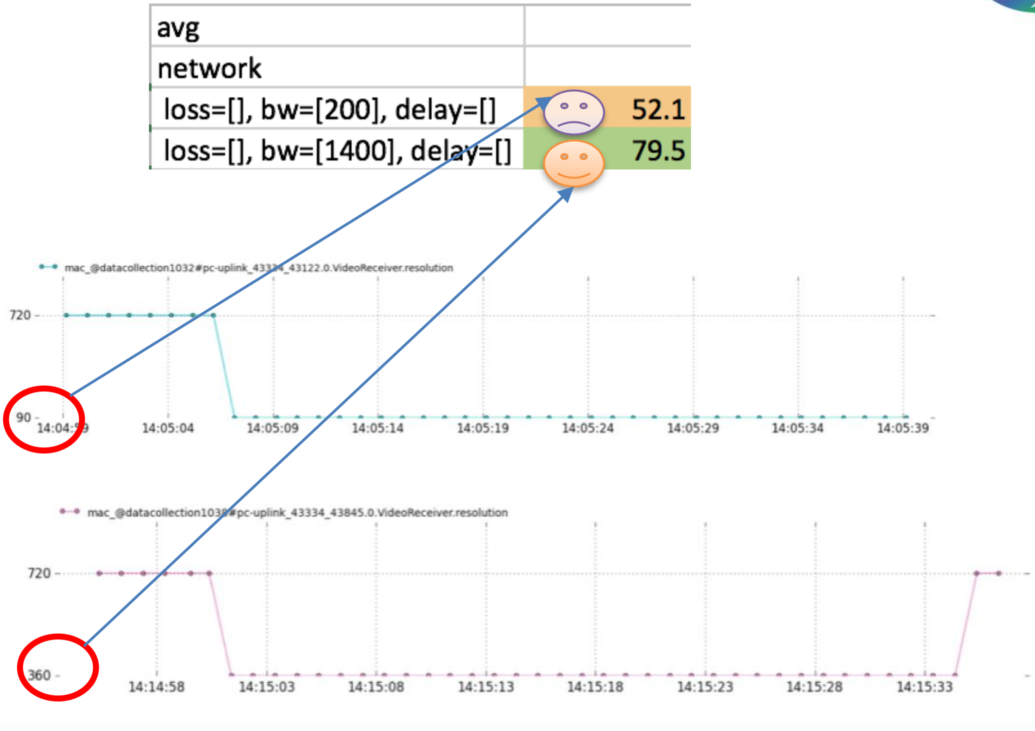
case	2018-01-31	2018-02-03	2018-02-08
1001_['qos=, loss=[], bw=[], delay=[]']	😐 58.4	🙂 83.2	😐 44.2







Signals



avg	
network	
loss=[], bw=[200], delay=[]	☹ 52.1
loss=[], bw=[1400], delay=[]	☺ 79.5

4 Summary and Looking Forward

Summary and Look forward

- Be practical, leverage all useful data
- Guard the quality along the software development/deployment cycle closely

- Better modeling
- Keep iteration
- Apply to more use cases
- Correlate signals with metrics
- Leverage AI

Thank you

LiveVideoStack
音视频技术社区

CSDN