Is my QA process successful?

"Decide it with Metrics"

Mesut Durukal

rapyuta robotics

mesutdurukal

DurukalMesut

mesutdurukal.com

November, 2020

Agenda

Introduction

Necessity

Benefits

Metrics

Optimization

Customization

Categories

Hazardous Metrics

Practical Application

Summary

Introduction

Why should we use metrics?

Answers to questions

- How long will it take to test?
- How much money will it take to test?
- How bad are the bugs?
- How many bugs found were fixed, reopened, closed, deferred?
- How many bugs did the test team did not find?
- How much of the software was tested?
- Will testing be done on time?
- Can the software be shipped on time?
- How effective were the tests?
- Are we using low-value test cases?
- What is the cost of testing?
- Was the test effort adequate?
- Could we have fit more testing in this release?

Further Implications

- Take decision for next phase of activities
- Evidence of the claim or prediction
- Understand the type of improvement required
- Take decision or process or technology change

Quality Metrics

Optimization of Metrics

How to select best to get an idea about success

- Passed test case percentage
- Failed test case percentage
- Blocked test case percentage
- Test Case Productivity
- Total number of test cases
- Number of tests run per time
- Test Design Efficiency
- Test Review Efficiency
- Average time to test a fix/feature
- Average time to test the full set
- Rework Ratio
- False Alarm Ratio
- Automation Rate
- Bugs found after release ratio
- Bug find rate
- Number of bugs per test
- Defects per Requirement
- Mean Time Between Failures (MTBF)
- Defect Removal Efficiency
- Fixed defects percentage
- Accepted defects percentage
- Distribution by age
- Critical Defects Percentage
- Defect Severity Index (DSI)
- Defect Resolution Time
- Requirement coverage percentage
- Test Cases by Requirement
- Requirements without Test Coverage
- Total allocated costs for testing
- Actual cost of testing
- Budget variance
- Schedule Variance
- Bugs found by tests ratio
- Bugs found by review ratio
- Defects rejected percentage
- Deferred defects percentage
- Execution coverage percentage

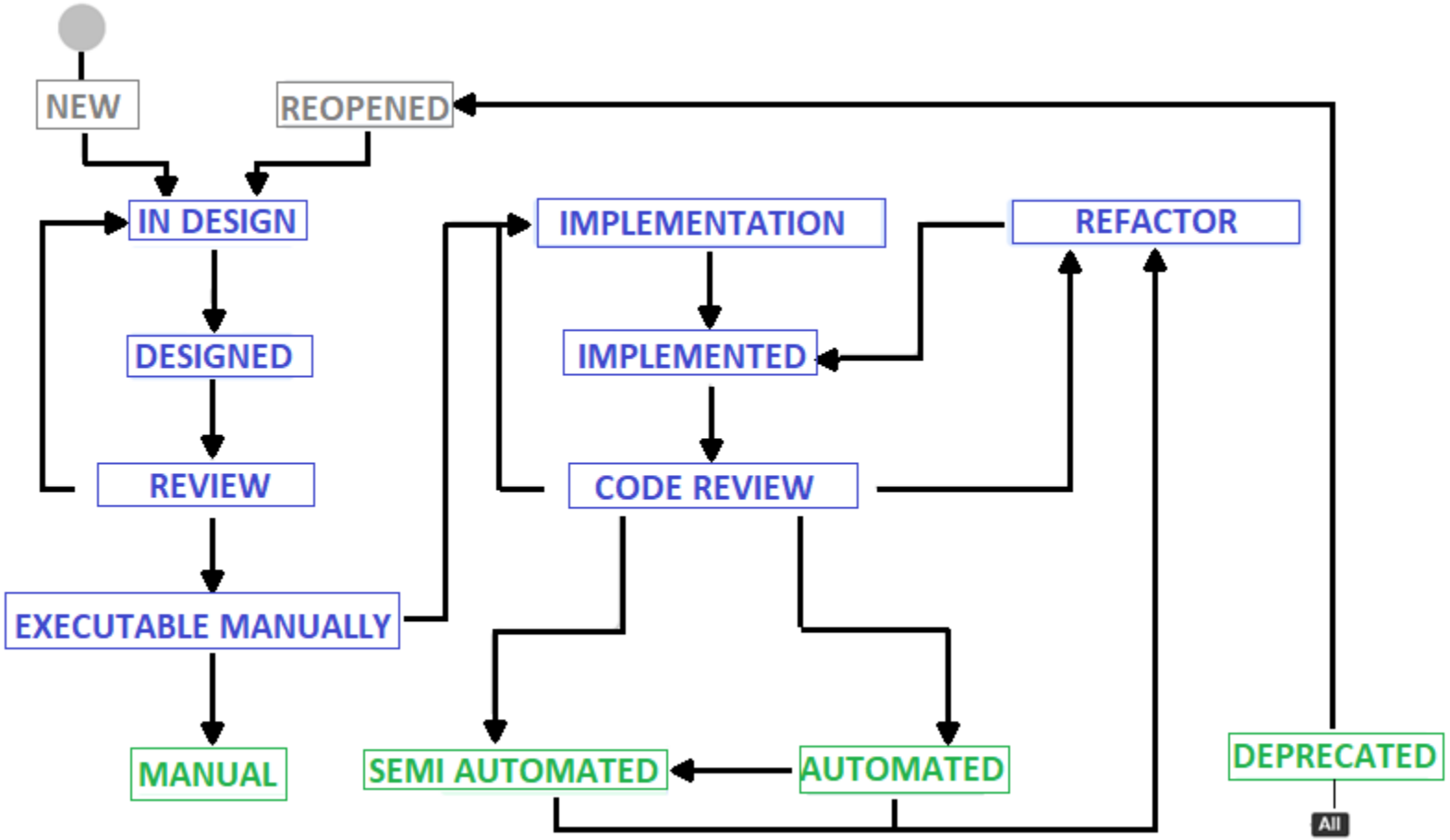
Did I choose the best metrics?

- Correlation
- Consistency
- Tracking
- Predictability
- Discriminative power
- Reliability

Cem Kaner

Customization

Time Elapsed for each step in Test Design



Customization

We add a custom field for tracking

```
<title>AWS Mindsphere JIRA</title>
<link>https://jira.mindsphere-tools.siemens.cloud</link>
<item>
    <title>[QAGP-59] EventManagement: Create custom events in a
    <link>https://jira.mindsphere-tools.siemens.cloud/browse/QA
    <project id="11105" key="QAGP">QA - Gone productive</projec
    <description><p>Create cutom event types as admin user from
    from them.</p> <p> </p> <p>Hint for implementation: should
    <key id="51963">QAGP-59</key>
    <summary>EventManagement: Create custom events in all possi
    <type id="10800" iconUrl="
    https://jira.mindsphere-tools.siemens.cloud/download/resour
    <priority id="10301" iconUrl="https://jira.mindsphere-tools
    </priority>
    <status id="12006" iconUrl="https://jira.mindsphere-tools.s
    case automated and ready to execute it as code / script.">A
    <statusCategory id="3" key="done" colorName="green"/>
    <resolution id="-1">Unresolved</resolution>
    <assignee username="Z003DDNF">D, Mesut</assignee>
    <reporter username="Z00365FU">VG, Christina</reporter>
    <customfields>
        <customfield id="customfield_15301" key="com.atlassian.
            <customfieldname>Average Execution Duration</custom
            <customfieldvalues>
                <customfieldvalue>18.0</customfieldvalue>
            </customfieldvalues>
        </customfield>
    </customfields>
</item>
```


```
[
  {
    "id": 1,
    "iid": 1,
    "project_id": 3,
    "title": "test1",
    "description": "fixed login page css paddin
    "state": "merged",
    "merged_by": {
      "id": 87854,
      "name": "Douwe Maan",
      "username": "DouweM"
    },
    "merged_at": "2018-09-07T11:16:17.520Z",
    "closed_by": null,
    "closed_at": null,
    "created_at": "2017-04-29T08:46:00Z",
    "updated_at": "2017-04-29T08:46:00Z",
    "target_branch": "master",
    "source_branch": "test1",
```

Time

- Average time to test a fix/feature
  - Blocking fast release: not success
- Average time to test the full set
- MTTF, MTBF
- Mean Defect Resolution Time
  - Defect distribution by age
- Schedule slippage

$$\frac{\text{(Actual end date – Estimated end date)}}{\text{(Planned End Date – Planned Start Date) X 100}}$$

Cost

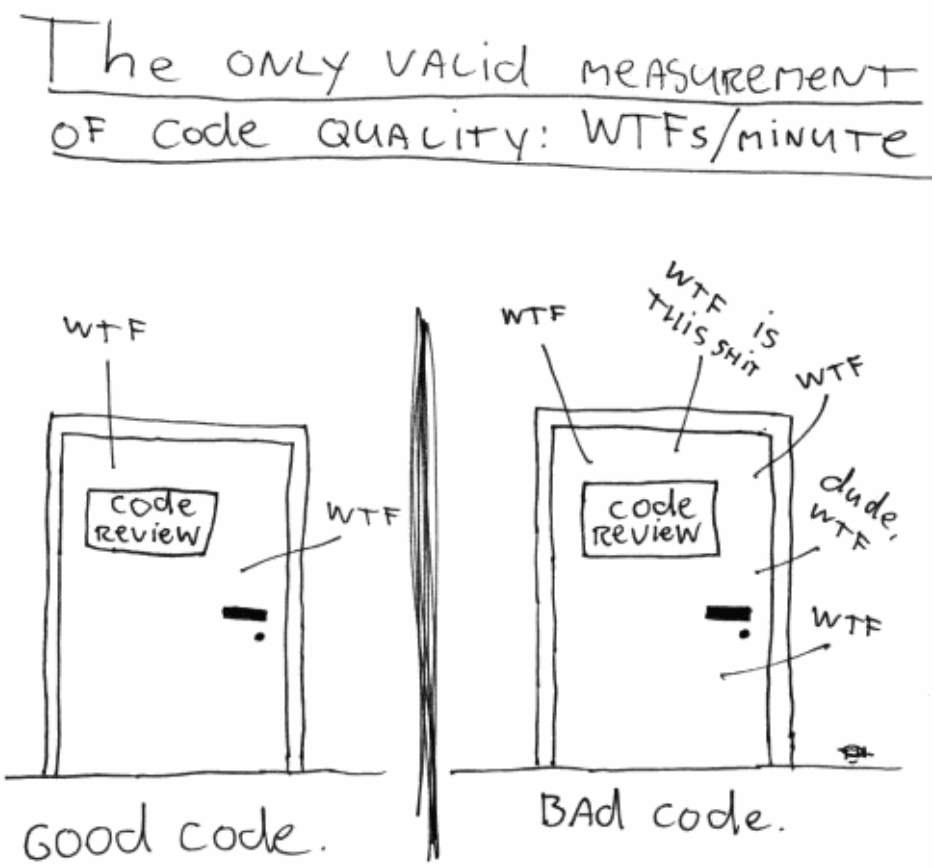
- Total allocated cost
  - How many people are you allocating for QA?
  - Other resources
- Execution costs
  - Too many executions: AWS cost. (Stability)
- Cost for tools/licenses
- Budget variance

Quality

- Test Cases Status (Passed/Failed/Blocked test case percentage)
- Defects per requirement
- Defects status (Fixed Ratio)
- Defects severity distribution (Critical / total)
  - Defect Severity Index (DSI) = (Defect * Severity Level) / Total defect number
- Non-functional aspects: performance, usability, compatibility

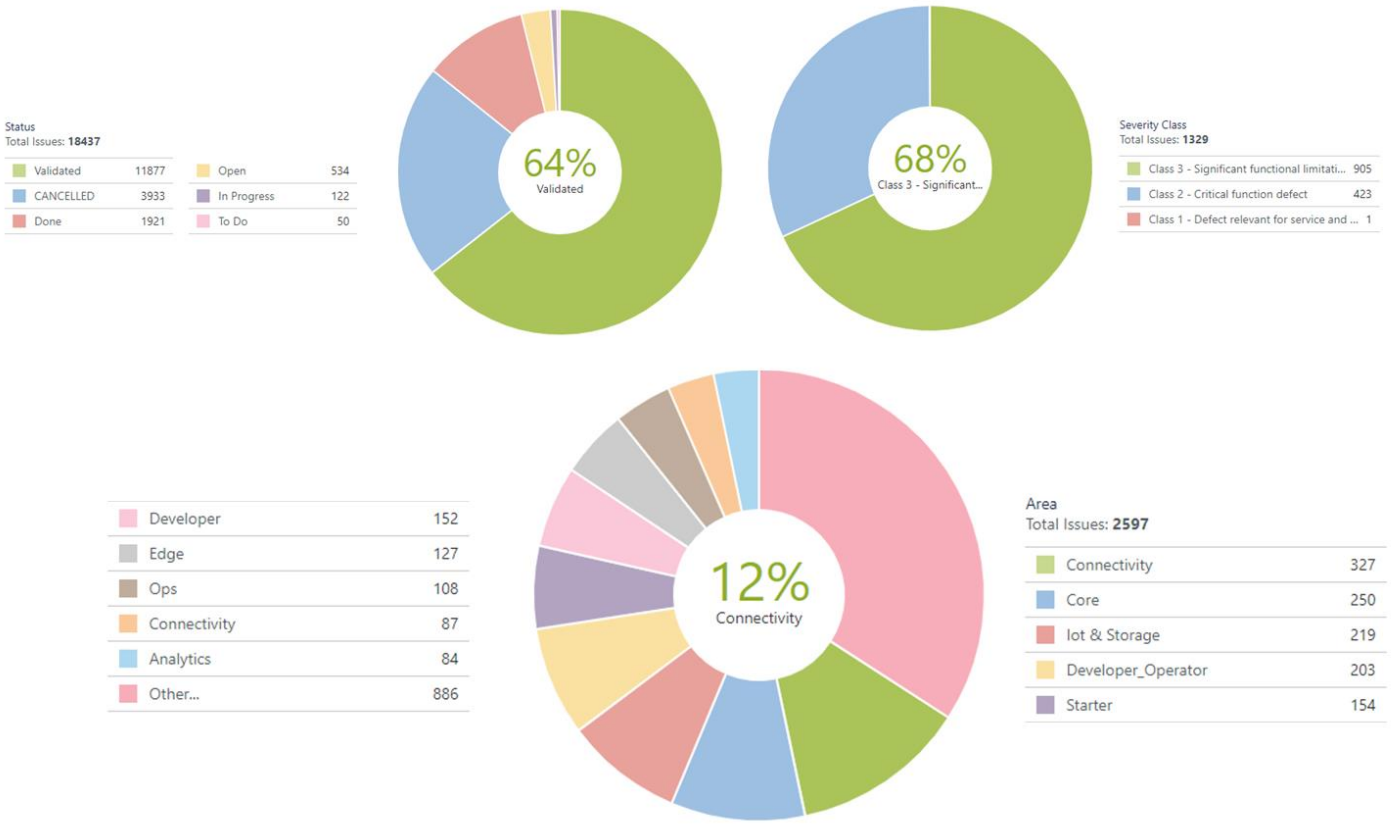
Inner Quality

- Method name length
- Method line length
- Code complexity
- Code coverage

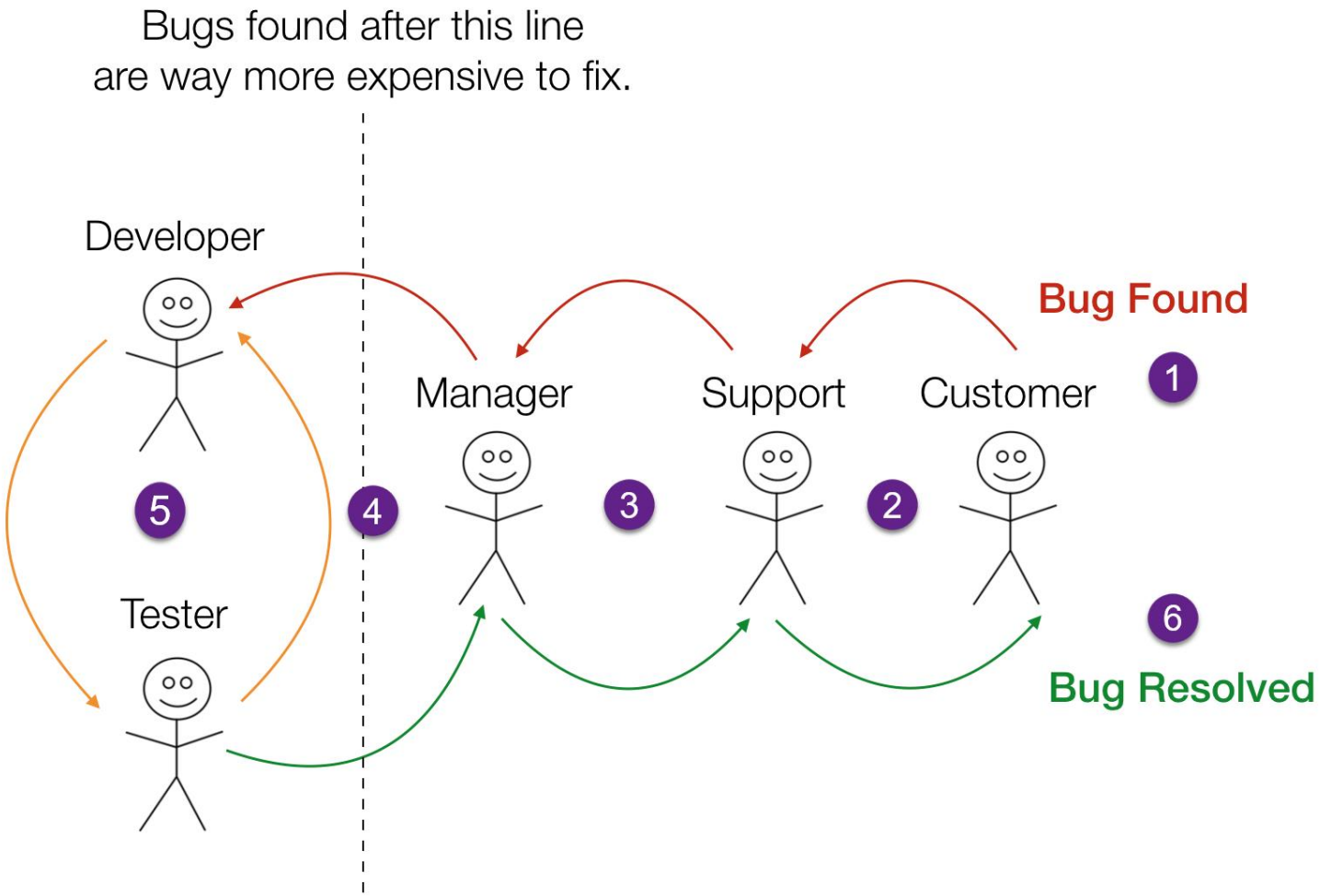


A Reference for Quality

Bugs



Escaped Bugs



by Dean Peters

Efficiency

- Productivity
  - Test Design Efficiency
    (nr of test designed / time)
  - Test Review Efficiency
    (nr of test reviewed / time)
  - Test execution efficiency
    (nr of test run / time)
  - Rework Effort Ratio
    (refactoring/new implementation)
  - False Alarm Ratio
- Maturity
  - Total number of test cases
  - Automation Rate
- Coverage
  - Test execution coverage
    (nr of test run / total test nr)
  - Requirement coverage
    (nr of requirements covered /
    total requirements)
  - Test case number per requirement
- Bugs
  - Distribution of bugs in stages
    (static code -> unit -> e2e -> customers!)
  - Density
    (total nr of defects / test case)
  - Defect acceptance rate
    (accepted / total)

This time:

Emotional Metrics

- Satisfaction
- Delight
- Pride
- Core Values
- Know-How
- Cultural Memory

Hazardous Metrics

- Story points
- Bug number
- Anything that is strict count

Practical Application

Practical Application

Automatically Collect Metrics over Jira Continuously

```
public HashMap<String, Long> measureDefectResolutionDurationPerArea() {
    String jql = "type = Bug AND status in (Validated) AND creator in " + reporterList + " and area = ";

    ArrayList<String> suiteList = getSuiteList();
    HashMap<String, Long> defectResolutionDurations = new HashMap<String, Long>();

    for (String suite : suiteList) {
        Promise<SearchResult> searchJqlPromise = jiraRestClient.getSearchClient().searchJql(jql + suite, 1000, 0,
                null);
        SearchResult claim = searchJqlPromise.claim();
        int totalNumberOfBugsPerSuite = claim.getTotal();

        long totalResolutionDurationInHoursPerArea = 0;
        long avgResolutionDuration = 0;
        if (totalNumberOfBugsPerSuite != 0) {
            for (Issue issue : claim.getIssues()) {
                DateTime creationDate = issue.getCreationDate();
                LocalDateTime creationDateInLocalDateTime = LocalDateTime
                        .parse(creationDate.toString().substring(0, 23), DATETIMEFORMATTER);
                long creationDateInSeconds = creationDateInLocalDateTime.toEpochSecond(ZoneOffset.UTC);

                String resolutionDateInString = (String) issue.getFieldByName("Resolved").getValue();
                LocalDateTime resolutionDateInLocalDateTime = LocalDateTime
                        .parse(resolutionDateInString.substring(0, 23), DATETIMEFORMATTER);
                long resolutionDateInSeconds = resolutionDateInLocalDateTime.toEpochSecond(ZoneOffset.UTC);

                long diffInSeconds = resolutionDateInSeconds - creationDateInSeconds;
                long diffInHours = diffInSeconds / 3600;

                totalResolutionDurationInHoursPerArea = totalResolutionDurationInHoursPerArea + diffInHours;
            }
```

"totalNumberOfBugsPerSuite"= 94

94

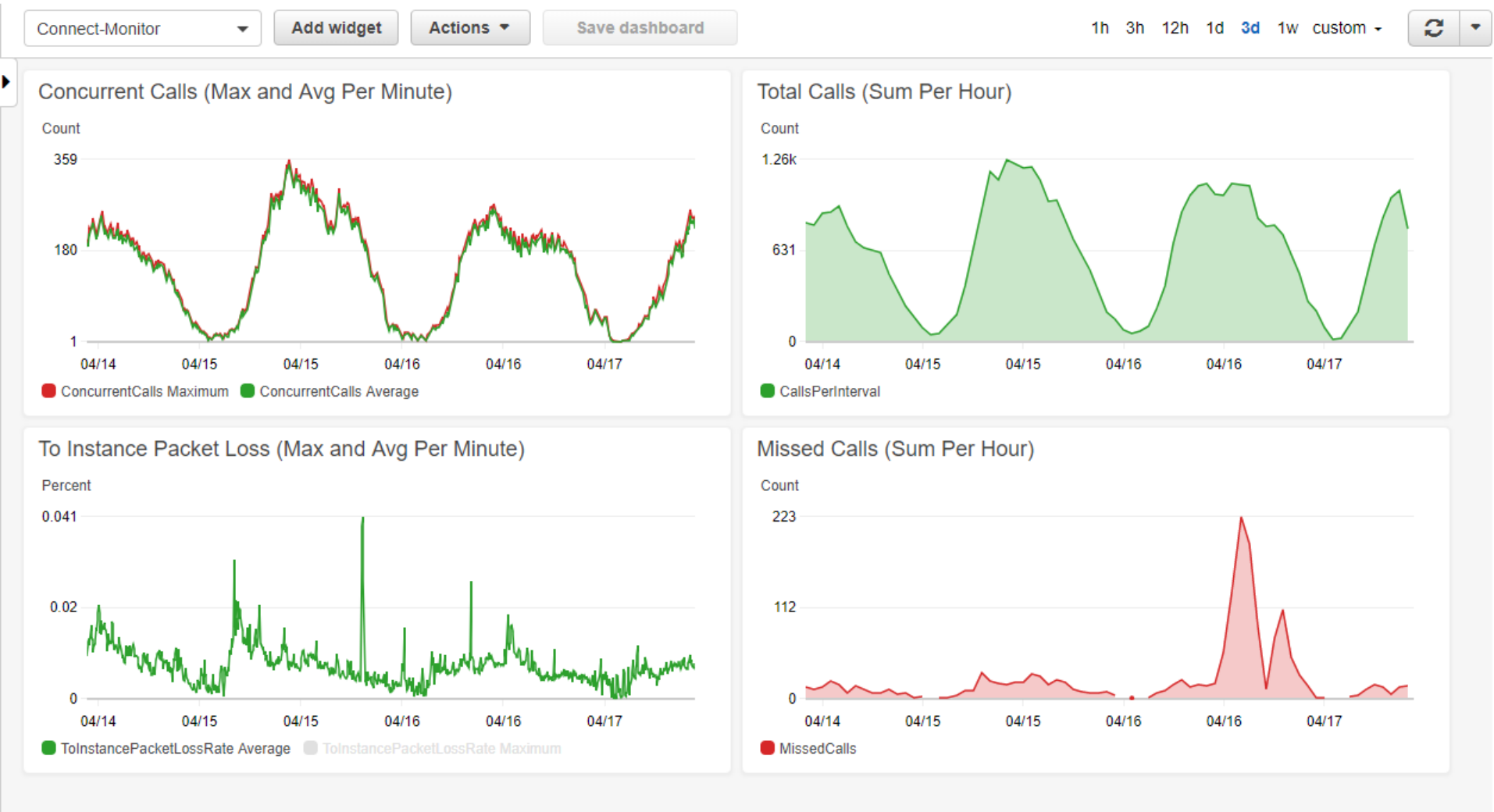
Practical Application

Not to lose any data: Upload all to Cloudwatch

```
logger.info("Read Credentials over config file and Connect");
AmazonCloudWatch cw = new CloudWatchHelper().getAwsCloudWatch();

HashMap<String, Integer> myMap = new HashMap<String, Integer>();
myMap.put("Alice", 52);
myMap.put("Bob", 53);
myMap.put("Mesut", 54);
myMap.put("Ahmet", 55);

for (Map.Entry<String, Integer> entry : myMap.entrySet()) {

    String key = entry.getKey();
    Double value = entry.getValue().doubleValue();

    Dimension dimension = new Dimension().withName("Children").withValue(key);
    MetricDatum datum = new MetricDatum().withMetricName("Their Notes").withUnit(StandardUnit.None)
            .withValue(value).withDimensions(dimension);

    PutMetricDataRequest request = new PutMetricDataRequest().withNamespace("Notes of Children in My Class")
            .withMetricData(datum);

    cw.putMetricData(request);

}
```

Practical Application

Use Dashboards for Monitoring: Grafana



CONCLUSION


NEWS
FORUM
SOCIAL NETWORK
CONTENT SHARING
MICROBLOGS
SOCIAL MEDIA
VIDEO
EMAIL
COMMUNICATION
CHAT
PICTURE
SMS
OUR BUSINESS
IDEA
YES
?
FIREWALL
SERVER FARM
INTERNET
COMPANY HEADQUARTERS
GROWTH
STATISTICS
LAST YEAR
THIS YEAR
35%
TIME
MONEY
BUSINESS
QUALITY
ROUTER
QUALITY
STRATEGY
VISION
SUCCESS
INNOVATION
VISION
RESEARCH
SOCIAL NETWORK

- Optimize
- Customize
- Categorize
- Hybridize
- Interpret


thank you
merci
danke
謝謝
gracias
teşekkür ederim
ngiyabonga
tapadh leat
dziękuję
bedankt
mauruuru
hvala
go raibh maith agat
sukriya
kop khun krap
terima kasih
dakujem
arigatô
takk
grazie
-taiku
sagolun
obrigado
спасибо
рахмат
Баярлалаа
dank je
misaotra
matondo
paldies
grazzi
mahalo
хвала
asante
obrigada
murakoze
chokrane
mamnun
дякую
mochchakkeram
djere dieuf
tau
akun
dankon
ačiū
spas
welalin
tack
barka
mersi
kia ora
vinaka
спасибі
blagodaram
faafetai lava
kiitos
dankie
nandri
nanni
dhanyavad
gràcie
bayarlalaa
köszönöm
chnorakaloutioun
gratias ago
gràcies
sulpáy
sobodi
děkuji
mèsi
didi madloba
najis tuke
kam sah hamnida
rahmat
rahmet
tanemirt
ありがとう
তোমাকে ধন্যবাদ
감사합니다
xièxie
ευχαριστώ
diolch
dhanyavadagalu
shukriya
merce
trugarez
chnorakaloutioun