

GIAC

全球互联网架构大会

GLOBAL INTERNET ARCHITECTURE CONFERENCE

使用代码度量提高代码质量

邱化峰 饿了么 测试开发专家



TOP100Summit

全球软件案例研究峰会

时间：11月15~17日

地点：北京国际会议中心

100个年度最值得学习案例

MPD工作坊（深圳站）

时间：9月21~22日

地点：深圳博林圣海伦酒店

20个3小时大时段沙盘课程

Make Professional
mpd

100

Make Professional
mpdMake Professional
mpd**MPD工作坊（北京站）**

时间：7月06~07日

地点：北京国家会议中心

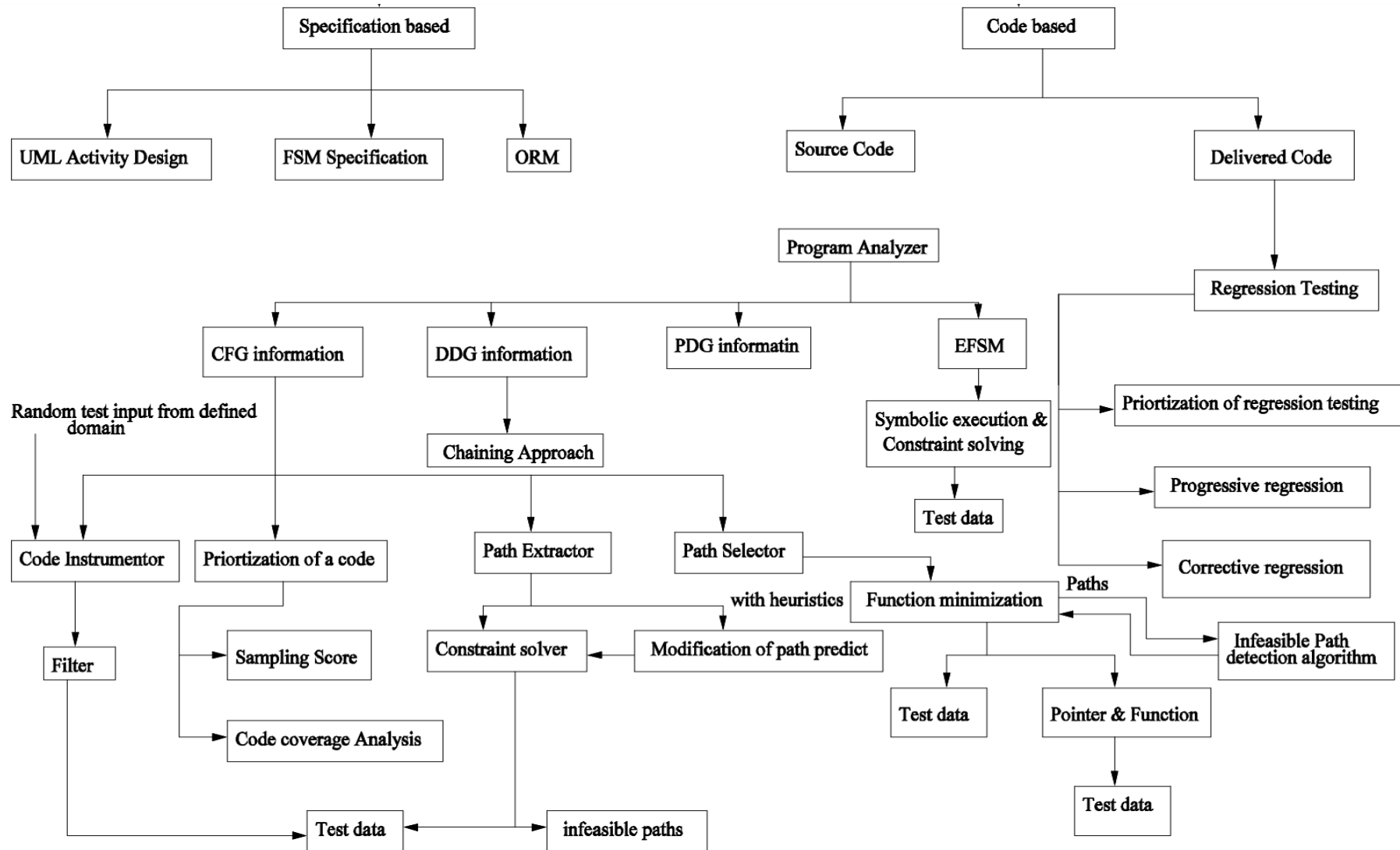
20个3小时大时段沙盘课程

MPD工作坊（上海站）

时间：10月26~27日

地点：上海

20个3小时大时段沙盘课程



一、度量指标-复杂度

圈复杂度(Cyclomatic complexity)是一种代码复杂度的衡量标准，在1976年由Thomas J. McCabe, Sr. 提出。

在软件测试的概念里，圈复杂度用来衡量一个模块判定结构的复杂程度，数量上表现为线性无关的路径条数，即合理的预防错误所需测试的最少路径条数。圈复杂度大说明程序代码可能质量低且难于测试和维护，根据经验，程序的可能错误和高的圈复杂度有着很大关系

其他度量方法包括：所用的数据量、控制结构中的嵌套层数、代码行数、对同一变量的先后引用之间的代码行数（跨度）、变量生存的代码行数（生存期），以及输入和输出的量

0-5: 子程序可能还不错

6-20: 得想办法简化子程序了

10+ 把子程序的某一部分拆分成另一个子程序并调用它。



一、度量指标-UML OO 内聚 和 耦合

1. 类内聚缺乏度
2. 耦合类数量
3. 直接子类数量
4. 类继承深度
5. 类封装度
6. 类扇出耦合度



一、度量指标-UML OO 内聚 和 耦合

1. C&K面向对象度量集, 以支持向量机分析方法为数学工具, 建立一种基于面向对象软件易发性故障预测模型
2. W. Li and S. Henry Object-oriented metrics that predict maintainability
3. 布兰德 定义的高层次的面相对象的度量指标
4. 费尔南德斯定义的内聚的指标SCOM
5. 达拉勒定义的路径连接的内聚指标PCCC



一、度量指标- COMPONENT

PSU-provided service component

RSU-required service component

IDC-interaction density of component

IIDC- incoming interaction density of component

OIDC- outgoing interaction density of component

PCUM- provided component utilization metrics

RCUM- required component utilization metrics

ICCM- interaction complexity of a component metrics

SICCM- sharing interaction complexity of component metrics

AICSCM- average interaction complexity of software component metrics



一、度量指标- COMPONENT

Munson & Khoshgoftaar extracted four factors: volume, control difficulty, modularity, and action

three metrics: control, system and volume

define the size of a code artifact: its number of lines (LOC), number of tokens (NOT), number of statements (NOS), etc

the component of size might be described as: $0.4 \cdot \text{LOC} + 0.7 \cdot \text{NOT} + 0.6 \cdot \text{GZIP}^{0.5} \cdot \text{NOS} + 0.1 \cdot \text{MCC}$.



二、细化指标-思考

1) 控制流程对代码复杂度的影响, if 和for 是否拥有相同的复杂度

2) 条件表达式对代码复杂度的影响

- A) 逻辑表达式个数对于复杂度的影响

```
sum = 0;
for (i=1; i<=100; i++)
    sum += i;
```

- B) 使用多个相互关联的表达式和使用单个组合的逻辑之间的复杂度的变化

- C) 单个表达式和嵌套表达式的相关复杂度

- D) 否定表达式

```
if(A || B){print(1)}
else{print(2)}
```

↔

```
if(A){print(1)}
else if(B){print(2)}
else{print(3)}
```

3) 编程风格对代码复杂的影响



二、细化指标-思考

```

f*:  for (var i=0 ; i<3 ; i++)
      if (x>10*2*i && x<10*(2*i+1)) { ... }

f[]:  var a = [[0,10] , [20,30] , [40,50]];
      for (var i=0 ; i<a.length ; i++)
        if (x>a[i][0] && x<a[i][1]) { ... }

```

个数越多越容易出错

组合的条件表达式可以转化为 嵌套的表达式, 但嵌套的表达式会增加行数的计算

```

al:  (x>0 && x<10 || x>20 && x<30 || x>40 && x<50)
an:  (x>0 && x<50 && !(x>10 && x<20) && !(x>30 && x<40))
an1: (! (x<0 || (x>10 && x<20) || (x>30 && x<40) || x>50))
an2: (!x<0 && !(x>10 && x<20) && !(x>30 && x<40) && !(x>50))

```



三、结论-复杂度

<i>RQ</i>	<i>description</i>	<i>results</i>
1	if vs. for	for loops are harder than ifs
2a	expression size	3 predicates is harder than 2 3 vs. 4 not significant
2b	compound vs. structure	differences not statistically significant
2c	flat vs. nesting	flat structures appear to be slightly easier
2d	negation	some but not all uses of negation are harder: negations are different from each other
3	loop idioms	loops counting up are easier unusual loop bounds do not have a significant effect

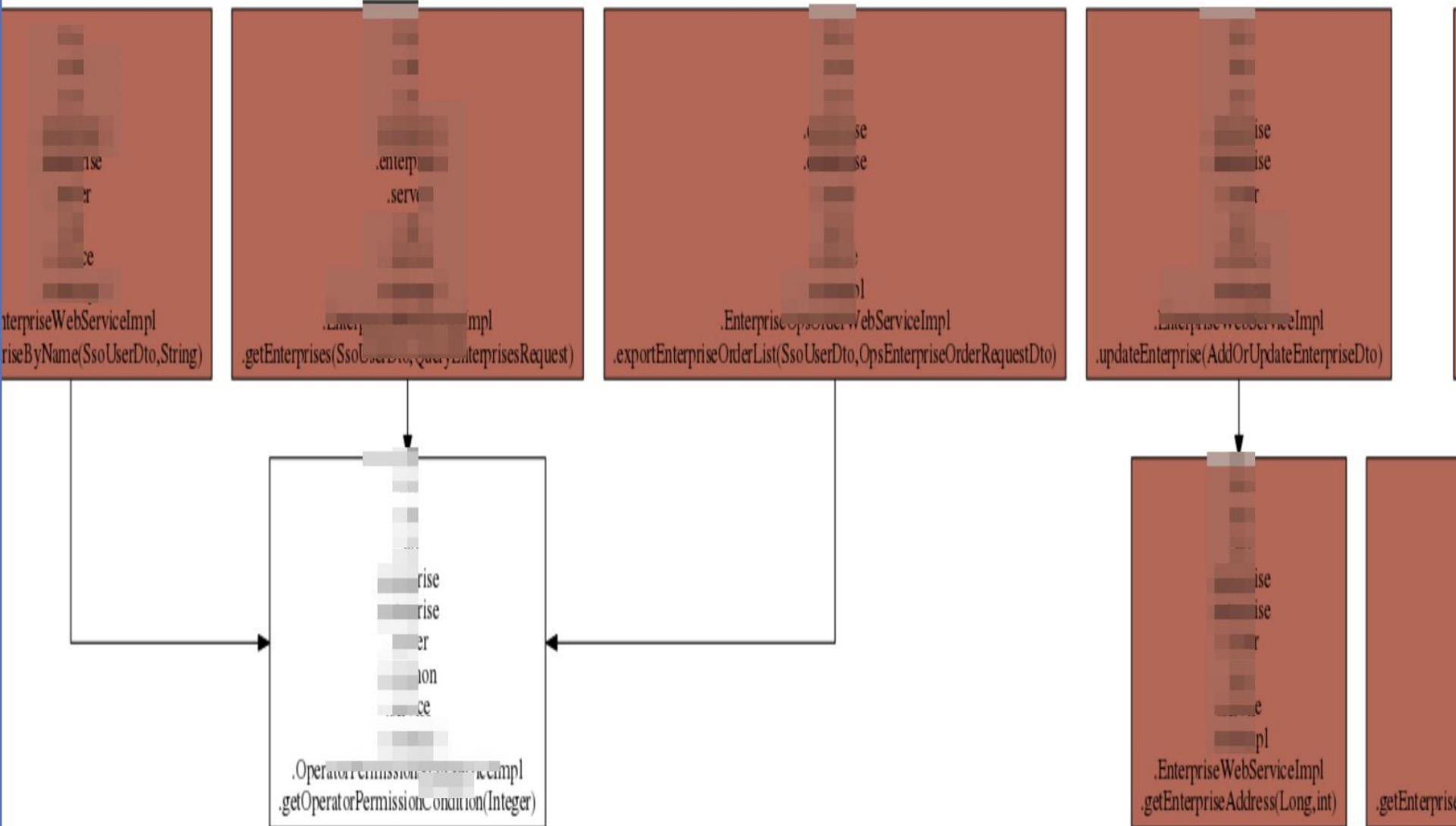


三、结论-复杂度

<i>RQ</i>	<i>description</i>	<i>results</i>
1	if vs. for	for loops are harder than ifs
2a	expression size	3 predicates is harder than 2 3 vs. 4 not significant
2b	compound vs. structure	differences not statistically significant
2c	flat vs. nesting	flat structures appear to be slightly easier
2d	negation	some but not all uses of negation are harder: negations are different from each other
3	loop idioms	loops counting up are easier unusual loop bounds do not have a significant effect



三、结论-链路的复杂度



三、结论-链路的复杂度

POST
https://atfcapi.alta.elenet.me//atfcapi/pictSOA/getDiffInterfaces
Params
Send
Save

Authorization
Headers (1)
Body
Pre-request Script
Tests
Cookies
Code

form-data
x-www-form-urlencoded
raw
binary
JSON (application/json)

```

1 {
2   "appid": "backend_service",
3   "env": "altb1",
4   "prodCommitId": "3883efc...f3...06ae848585babbfaff5c",
5   "commitId": "e150c1086...h44edce...6b1baacc44af2690d"
6 }

```

Body
Cookies
Headers (9)
Test Results

Status: 200 OK
Time: 56214 ms
Size: 1.42 KB

Pretty
Raw
Preview
JSON

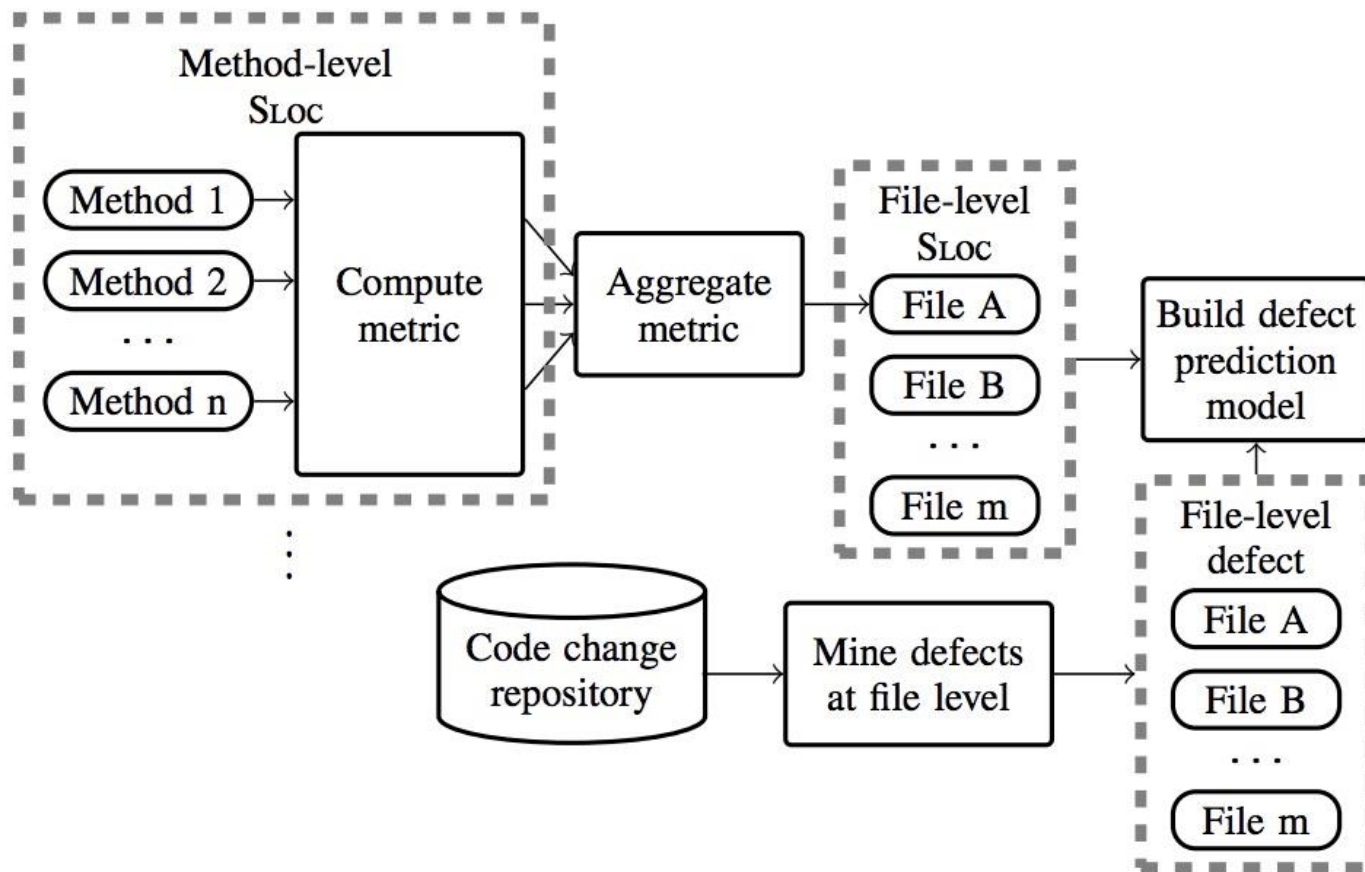
Save Response

```

1 {
2   "code": "200",
3   "msg": "OK",
4   "data": [
5     "eMakeUpTaskService.execute",
6     "CommodityCentreWebService.saveCommodityLocation",
7     "CommodityCentreWebService.signNew",
8     "pTaskService.makeUpByShelfGroup",
9     "pTaskService.setExpireTimeSaleInfo",
10    "queryCommodityInfo",
11    ".operationExpireTime",
12    "akeUpTaskService.saveLog",
13    "CommodityCentreWebService.queryCommodityLocation",
14    "akeUpTaskService.main",
15    "imeUpTaskService.DB",

```

三、结论-链路的复杂度-缺陷预防



三、结论-基于Metric的统计结果

Metric 统计

1.Kendall rank correlation coefficient (皮尔森)

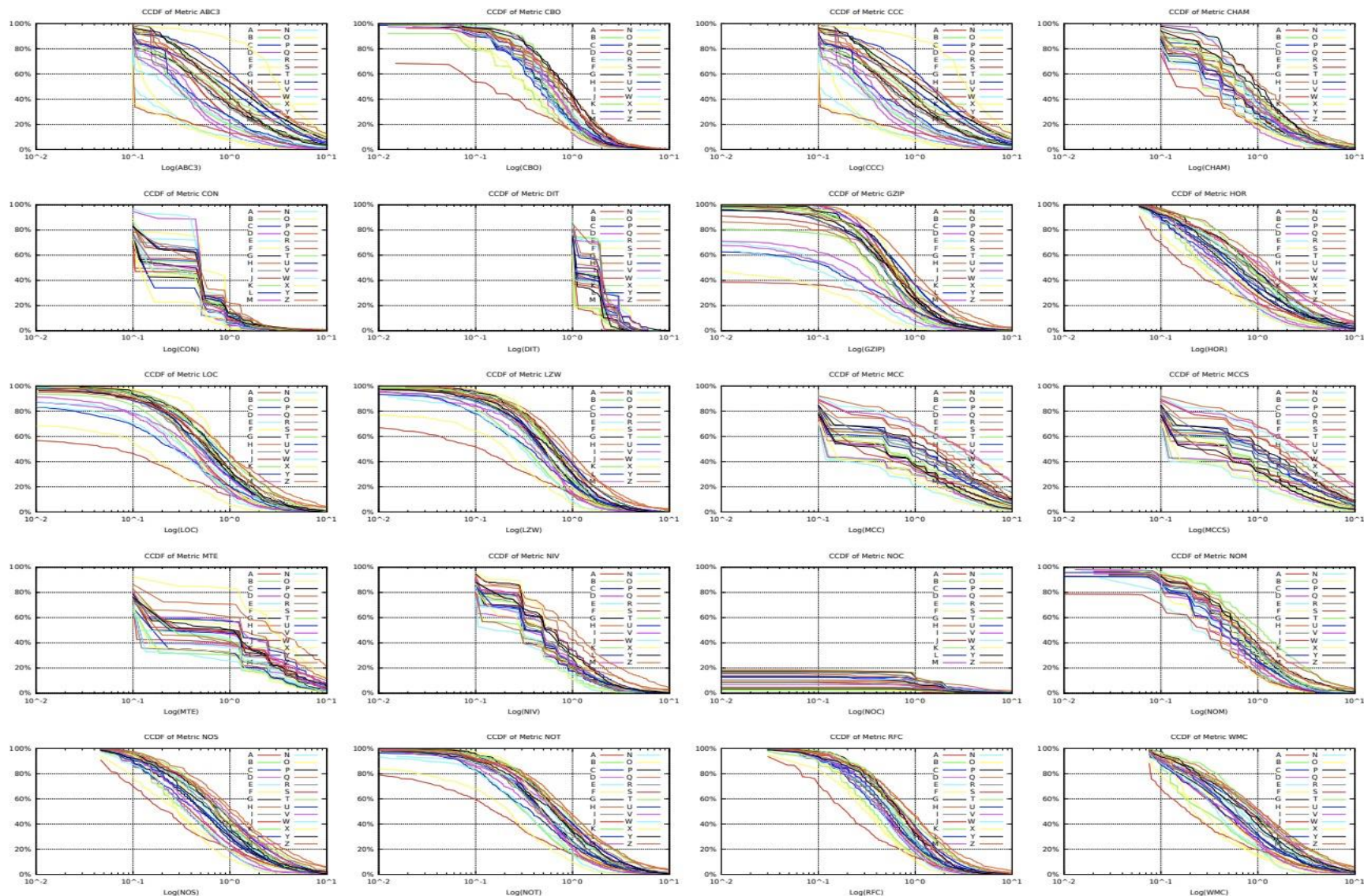
2.Correlation between the metrics

3.线性分布

4. 优先级



三、结论-基于Metric的统计结果





欢迎关注msup微信公众账号

关注大会微信公共账号，及时了解大会动态、
日程及每日更新的案例！

