



**中国软件测试认证委员会**  
**Chinese Software Testing Qualifications Board**

## 采用精准测试技术在云端实施 可信、可视化软件测试

演讲人：赵明

1 Page 05-07 互联网+“软件测试”模式分析

2 Page 08-15 精准测试对测试公信力的技术创新

3 Page 16-22 可视化测试系统

4 Page 23-28 云端原生测试管理系统

5 Page 29-32 计算机辅助测试分析技术

6 Page 33-35 在“云端”运行的白盒测试技术

7 Page 36-39 企业级技术特性

### 传统测试方法互联网化遇到的问题

#### 瓶颈

沿用普通黑盒测试方法，在测试后期检测效率极低，无法高效检出缺陷。互联网测试环境下该问题更加突出。

低效

#### 瓶颈

记录在测试管理系统中的测试数据都完全依赖人工录入数据，几乎没有公信力。无法在没有高强度管理的互联网环境运行。

可信性

#### 瓶颈

所有追踪结果基于传统“软件测试”元素层面，并无法从测试覆盖层面展开精确的量化和定性。

量化

### 互联网众测模式分析



基于软件测试=发现bug的假设

有违软件测试工作的基本工作原则

考核  
标准

商业  
模式

众测人员测试的大量重复用例使得难以商业模式模糊  
参与者无组织的测试导致用例高度重复，摊薄参与者的  
经济回馈使得众测平台参与度低

非专业人员的介入和互联网模式使得缺陷确认消耗大量时间

表面大量低成本的测试资源，由于需要对产生的“不规范的、不  
专业化、”结果进行审核，综合成本并无法有效降低。

时间  
成本

### 基于互联网的测试协同非常困难

#### 项目研发团队 (代码为工作核心)

- 在互联网端发现的缺陷，描述、复现和Debug难度极大。
- 测试成果的可见性在互联网上更加模糊，产品研发团队完全无法看到测试的成效

- 互联网端进行的测试，测试人员得到的程序逻辑描述更加模糊
- 对于变更的把握将更加模糊，在版本迭代、回归测试中将存在很大的测试风险。

#### 互联网测试团队 (测试用例为工作核心)

### 精准测试的介绍



采用专业的测试软件，对软件测试执行全过程的原生数据进行自动采集、存储、运算、可视化展示，它依靠一系列的分析算法，可对软件测试的效率、质量进行分析、改进和优化。



### 在线精准测试的核心特性

- 1、测试工程师依然采用主流的黑盒测试方法，由计算机程序去采集程序执行逻辑以及其他测试数据的过程，只需要面对应用进行，无需代码。
- 2、所有数据由系统自动、原生录入，无法人工修改，保证数据精准和不可篡改性，精准的测试数据可直接用于测试的过程管理和实效分析。
- 3、支持测试数据的精准度量以及全面的、多维度的测试分析算法。  
将  
白盒测试的视角从覆盖率扩展到计算机辅助测试分析领域。
- 4、基于迭代中的测试用例和代码的映射关系的，支持回归测试用例的自动选取。



### 精准测试

#### 自动识别黑盒-精准测试的换挡点

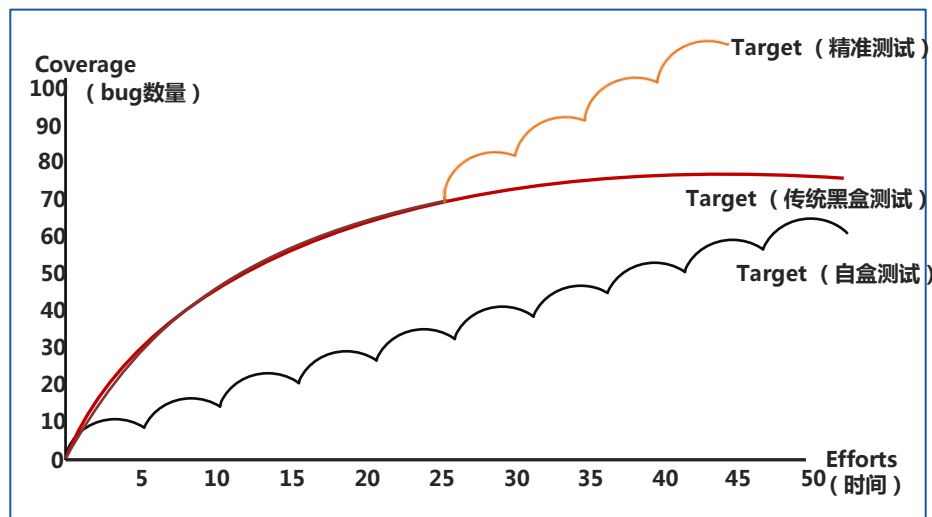
- 兼容传统测试模式，当手工黑盒测试效率进入瓶颈区后，引导用户转入精准测试阶段。

#### 全程测试可视化

- 可视化测试，每个测试用例的逻辑都可以单独映射到代码层，基于云进行测试的数据汇总和分析，并且引入智能算法对测试进行改进。

#### 彻底解决高复杂度、高速迭代，高可靠性产品级软件测试面临的矛盾。

- 通过内置强大的智能计算机辅助测试分析系统，帮助企业能够高效率完成复杂系统的测试。



### 可信的、精准软件质量管理

- 所有的数据录入，采用“软件示波器”，执行用例时对应的程序运行逻辑会自动、实时采集和分析。
- 所有数据精准到每个工程师、每个测试用例，并保证精准、无损、可追溯。
- 示波器的数据会真实反应测试的现场运行的代码层逻辑，通过专用接口自动输入数据，无法人工篡改。保证测试过程所有输入数据的精准性。



# 软件示波器：无线、自动采集 被测应用的代码运行逻辑和指标数据

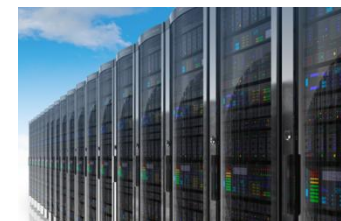


选择用例点击开始，然后对被测应用执行测试用例，对应的用例的代码数据就会被示波器自动采集。

## 软件示波器的优点：

- 辅助测试人员对被测试程序进行分析，例如等价类的辅助划分、探索性测试等。
- 通过程序实时输出的逻辑单元运行的数量，分析程序行为以及潜在的性能问题。
- 可随时重新录制，更新数据。

### 被测应用与示波器的通信方式





### 测试用例与代码的自动关联关系建立



- 将测试用例和海量的代码执行信息自动关联。
- 用户可以通过相关视图实现测试用例到代码的双向关联分析。

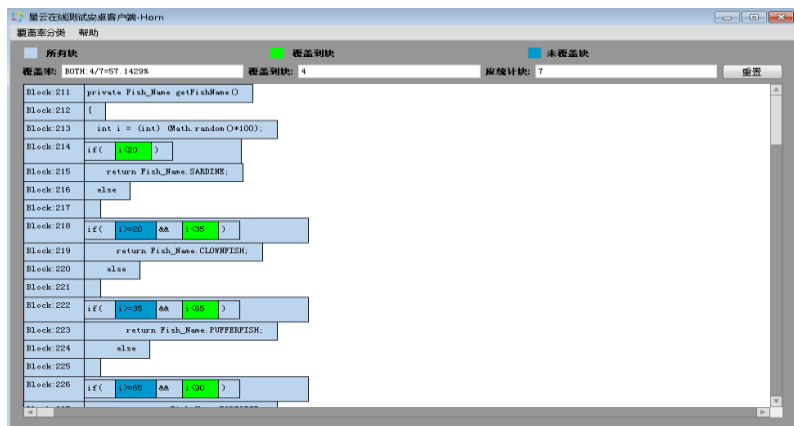
# 精准测试技术帮助建立代码维护资源库

**基于精准测试技术，测试过程将为客户系统保存关键数据**

- 通过在测试过程中产生的测试数据，完整的展示每个功能点的实现逻辑，从代码结构，函数路径，条件执行，代码块覆盖等多级角度展示程序功能点;
- 精准分析每个代码段被哪些用例所执行，在大型软件的修改过程中，可以依赖此数据进行一致性的维护和修改



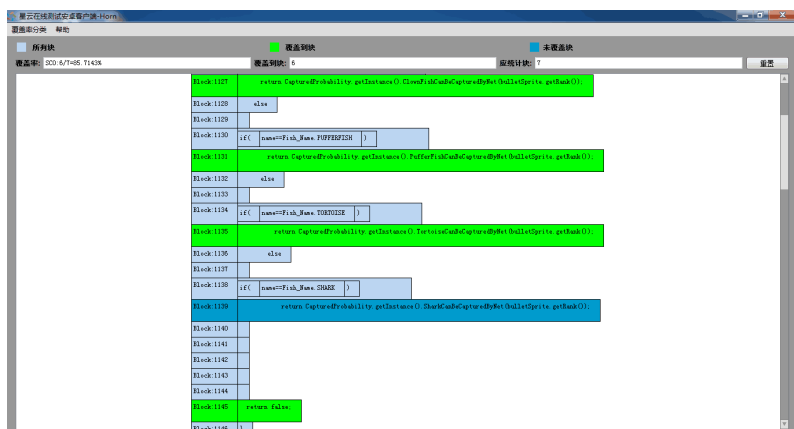
### 覆盖率计算可视化



➤ 支持对各种覆盖率（最高可达到MC/DC）的计算的可视化，对每种覆盖率的计算结果给出可视化的直观计算过程。

➤ 选择不同的覆盖率类型，未覆盖/已覆盖元素会自动切换。

➤ 给出每种覆盖率的分子/分母以及计算过程。





### 随机故障捕获：捕获所有闪退， 无需重现，极大减少定位问题时间



自动记录崩溃发生时刻之前程序执行的详细路径信息，捕获难以重现的缺陷并快速解决：

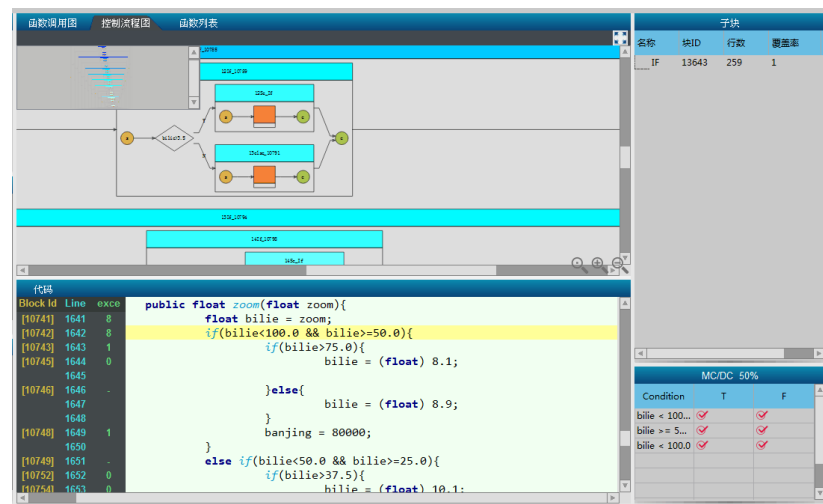
- 最后50个代码块。
- 最后50个条件。
- 最后50个判定执行。

除了在研发环境内，也可以基于云在用户现场精准定位缺陷。

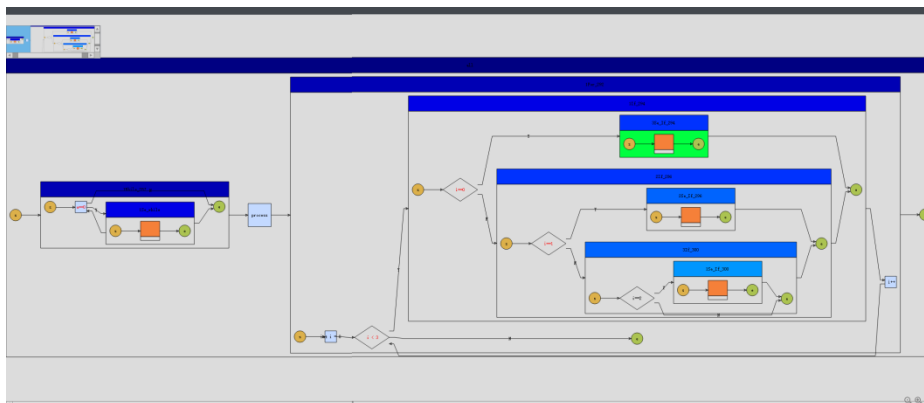
# 建立产品研发与互联网端测试建立可视化沟通桥梁

- 可自动记录互联网端每个测试缺陷的运行路径，快速定位并帮助研发团队精确定位缺陷并进行快速修复。

- 研发团队通过精准测试系统可视化测试运行的充分性查看漏测的代码逻辑，帮助互联网测试、开发进行用例审核和补充。



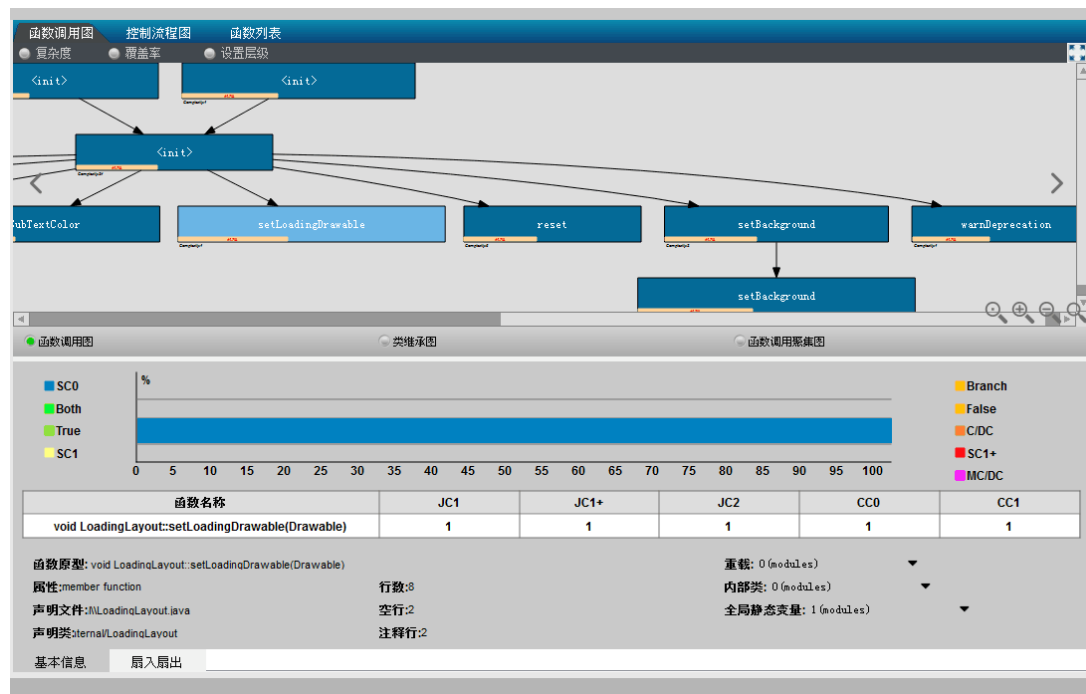
### 探索函数逻辑 “测试地图”



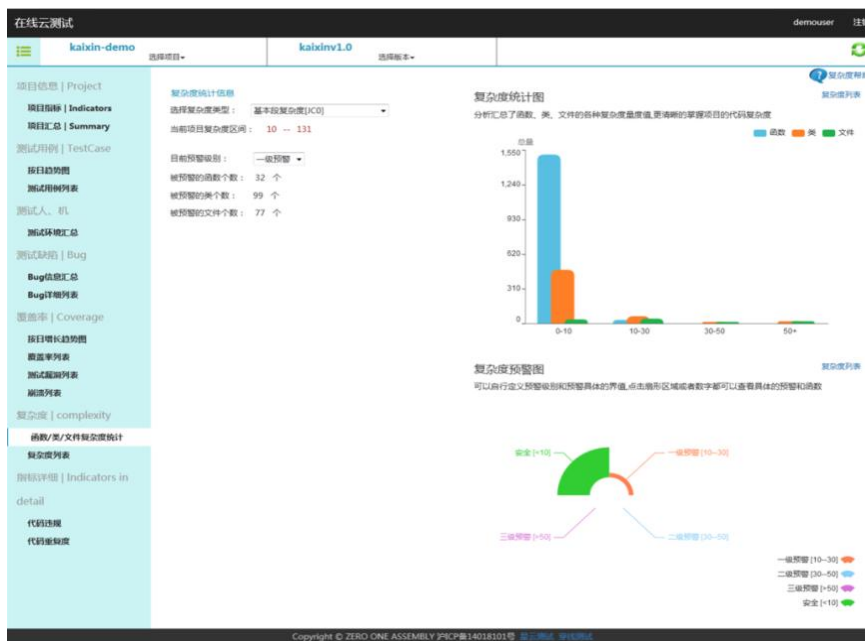
- 根据函数的内部控制逻辑，进行“测试地图”布局，具有实际意义的顺序块，以方块形式进行展示，并根据颜色区分动态覆盖信息。
- 通过对于该视图，可以清晰看到复杂模块的测试深度以及测试分布。
- 控制流程图可以与代码联动，选择相关节点及自动跳转到对应的代码位置，并且查看代码执行的动态信息。

### 探索函数调用上下文测试数据

- 动态执行信息可结合静态结构视图进行展示，程序总体运行情况一目了然。
- 在程序调用上下文中查看动态覆盖率，可以探索式发现程序潜在问题。
- 客户端视图可以全屏、拖动放大、缩小，帮助用户从全局分析程序结构。



### 静态复杂度分布视图



函数列表

默认只展示复杂度最高函数中个参考,请在选项设置中勾选复杂度,点击某一列会在列表下方绘制对应的雷达图

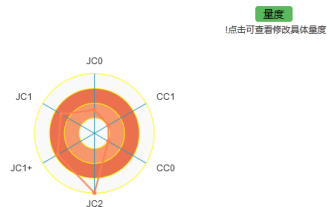
设置一页显示的个数: 10

共1541条记录

NO	函数 ID	函数名	所在类	JC0	JC1	JC1+	JC2	CC0	CC1
921	921	setListener	com/kaixin/android/menu/User	62	69	68	73	1	1
556	556	setListener	com/kaixin/android/activity/VoiceActivity	54	66	63	74	1	1
237	237	setListener	com/kaixin/android/activity/ImageFilterFaceAct...	46	48	48	50	1	1
806	806	setListener	com/kaixin/android/menu/Home	39	42	41	44	1	1
190	190	setListener	com/kaixin/android/activity/ImageFilterActivity	36	43	36	43	1	1
722	722	setListener	com/kaixin/android/menu/Desktop	35	38	37	40	1	1
263	263	setListener	com/kaixin/android/activity/ImageFilterFrameA...	33	34	33	34	1	1
380	380	setListener	com/kaixin/android/activity/PhotoPictureDetail...	30	37	30	39	1	1
1522	1522	combinatFrame	com/kaixin/android/utis/PhotoUtil	29	48	53	74	20	20
618	618	setListener	com/kaixin/android/activity/WriteRecordActivity	27	30	27	30	1	1

< 1 2 3 4 5 6 7 8 - 154 155 >

函数对应的雷达图



复杂度种类	最小值[min]	最大值[max]
JC0	0	40
JC1	0	40
JC1+	0	40
JC2	0	40
CC0	0	40
CC1	0	40

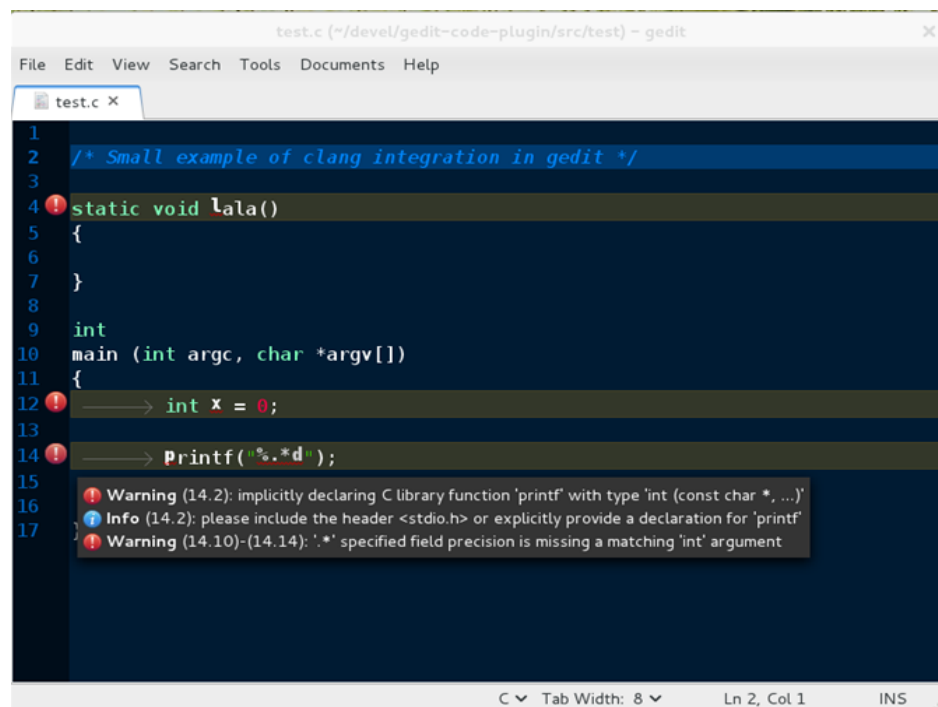
关闭 确定

➤ 确定工程中函数的复杂度分布。

➤ 通过雷达图分析每个函数的各种复杂度的符合性标准。

### 代码静态检查

- 直接对编译阶段语法树进行遍历匹配，匹配存在风险的代码
- 支持Cert，CWE以及商业工具fortify等权威机构定义的规则集
- 支持符号执行的检测器。
- 支持检测结果和代码的关联展示。



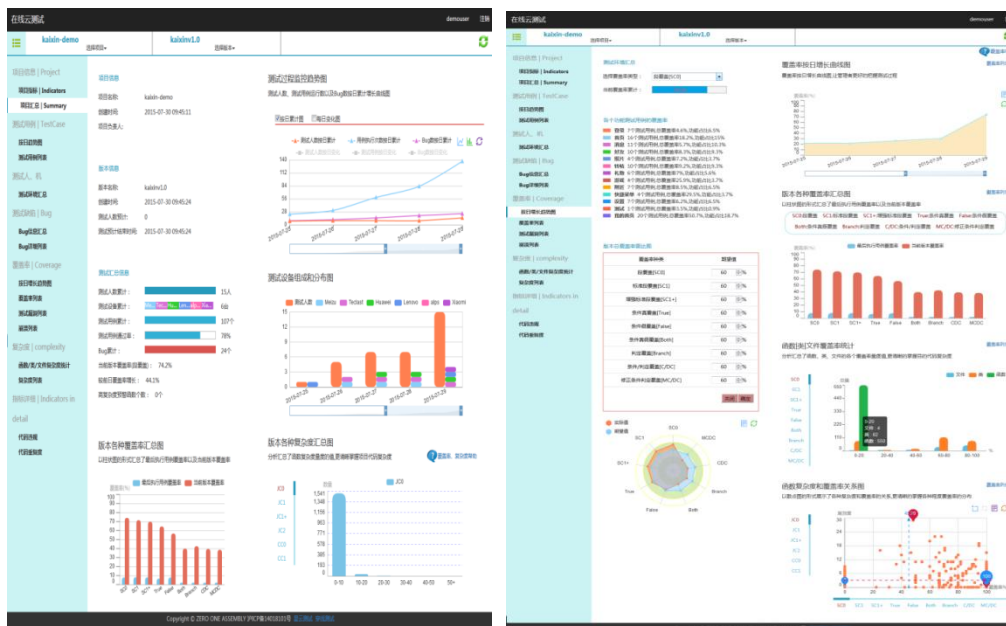
The screenshot shows a code editor window titled 'test.c (~/.dev/gedit-code-plugin/src/test) - gedit'. The code is a C program with the following content:

```
1
2 /* Small example of clang integration in gedit */
3
4 static void lala()
5 {
6
7 }
8
9 int
10 main (int argc, char *argv[])
11 {
12     int x = 0;
13
14     printf( "%.d" );
15
16
17 }
```

The editor displays several warnings and information messages in a panel at the bottom:

- Warning (14.2): implicitly declaring C library function 'printf' with type 'int (const char \*, ...)'
- Info (14.2): please include the header <stdio.h> or explicitly provide a declaration for 'printf'
- Warning (14.10)-(14.14): '\*' specified field precision is missing a matching 'int' argument

## 云端实时、精准质量跟踪



(测试客户端+后台实时Web追踪)

- 实时的web端测试报表。
- 云端管理系统展示的数据基于精准测试数据的分析，所有数据原生，精确，支持在云端进行互联网+测试。
- 测试团队、开发团队，甲方负责人等多种角色都可以登录系统，从各个层面对测试、软件质量进行分析。



## 云端精准质量趋势分析

覆盖率按日增长曲线图

覆盖率按日增长曲线图,让管理者更好的把握测试过程



覆盖率列表

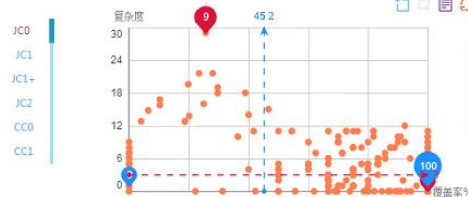
(黑盒效率换挡点+测试深度运动趋势)

- 累计覆盖率视图直观分析每天的工作实效以及确定从实施黑盒测试转换到精准测试的最佳时间点。

- 覆盖率和复杂度关系图,看到函数分布所处的位置和运动趋势,可以整体看到测试的深度和充分度情况。

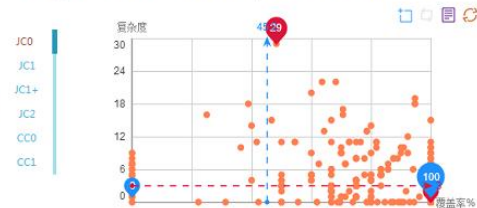
函数复杂度和覆盖率关系图

以散点图的形式展示了各种复杂度和覆盖率的关系,更清晰的掌握各种程度覆盖率的分布



函数复杂度和覆盖率关系图

以散点图的形式展示了各种复杂度和覆盖率的关系,更清晰的掌握各种程度覆盖率的分布





## 互联网端测试工程师实效精准分析系统

- 将参与的测试工程师的所执行的用例从逻辑覆盖映射到代码覆盖，真实记录并分析每个互联网测试参与者的工作实效。
- 以逻辑覆盖为基准的而不是用例数量为考核标准



测试用例排行图

测试用例列表

对于测试工程师利用t景制的测试用例个数累计给出排行榜,快速的找到表现突出的工程师

Pass Fail NotRun

共15条记录

排名	用例量	所属工程师编号	工程师姓名	贡献度
1	<div><div></div></div>	79	测试a部014	38.6%
2	<div><div></div></div>	15	测试a部006	30.5%
3	<div><div></div></div>	10	测试a部001	26.6%
4	<div><div></div></div>	76	测试a部011	24.5%
5	<div><div></div></div>	14	测试a部005	23.7%
6	<div><div></div></div>	11	测试a部002	22.4%
7	<div><div></div></div>	77	测试a部012	16.5%
8	<div><div></div></div>	13	测试a部004	12.3%
9	<div><div></div></div>	12	测试a部003	10.9%
10	<div><div></div></div>	16	测试a部007	10.6%

Showing: 1 to 10 Total: 15 rows 10 records per page

<< < 1 2 > >>

## 精确的测试用例执行场景信息记录

共108条记录

NO.	测试用例ID	测试用例名称	创建时间	执行时间	执行的设备	关联的bug	关联的函数	覆盖率占比	通过的状态	测试人
1	128	菜单消息发布	2015-07-28 16:32:09	2015-07-29 12:22:41	T8951_HUAWEI T8951	0	301	18.5	Pass	测试a部006
2	129	菜单位置发布	2015-07-28 16:32:30	2015-07-29 12:32:07	T8951_HUAWEI T8951	1	128	8.5	Error	测试a部006
3	131	导入好友	2015-07-28 16:53:42	2015-07-29 12:56:30	MSM8226_HM 1SW	0	96	4.9	Pass	测试a部007
4	132	查看上传进度	2015-07-28 16:55:42	2015-07-29 12:57:21	MSM8226_HM 1SW	0	27	1.6	Pass	测试a部007
5	133	打开设置	2015-07-28 16:57:06	2015-07-29 13:01:16	MSM8226_HM 1SW	0	96	5.4	Pass	测试a部008
6	134	用户反馈查看	2015-07-28 17:07:12	2015-07-29 12:57:57	MSM8226_HM 1SW	0	27	1.6	Pass	测试a部007
7	135	用户反馈提交	2015-07-28 17:09:44	2015-07-29 12:58:49	MSM8226_HM 1SW	1	14	0.4	Error	测试a部007
8	152	查看访客	2015-07-29 13:29:31	2015-07-29 17:00:48	MSM8226_HM 1SW	0	375	19.4	Pass	测试a部011
9	153	查看我的动态	2015-07-29 13:35:10	2015-07-29 17:16:50	teclast_P89s mini(F8A2)	0	82	4.4	Pass	测试a部011
10	154	点击关于	2015-07-29 14:50:06	2015-07-29 17:22:18	teclast_P89s mini(F8A2)	0	163	7.9	Pass	测试a部012
11	155	点击照片	2015-07-29 14:51:08	2015-07-29 17:23:10	teclast_P89s mini(F8A2)	0	196	11.3	Pass	测试a部012
12	156	点击日记	2015-07-29 14:51:40	2015-07-29 17:31:36	teclast_P89s mini(F8A2)	0	28	1.7	Pass	测试a部013
13	157	查看日记	2015-07-29 14:52:41	2015-07-29 17:31:57	teclast_P89s mini(F8A2)	0	38	2	Pass	测试a部013
14	158	我的日记刷新	2015-07-29 14:53:30	2015-07-29 17:32:25	teclast_P89s mini(F8A2)	0	37	1.5	Pass	测试a部013
15	159	编写日记	2015-07-29 14:54:51	2015-07-29 18:08:44	teclast_P89s mini(F8A2)	0	200	10.1	Pass	测试a部014

- 测试用例与测试覆盖率、测试设备、测试人员、测试时间等信息关联。
- 测试用例信息由系统自动采集和录入，可以保证信息的精确。

## Bug分析

### Bug关联报表：

测试设备与BUG的关联、

显示缺陷的增长以及与测试设备的分布关联关系

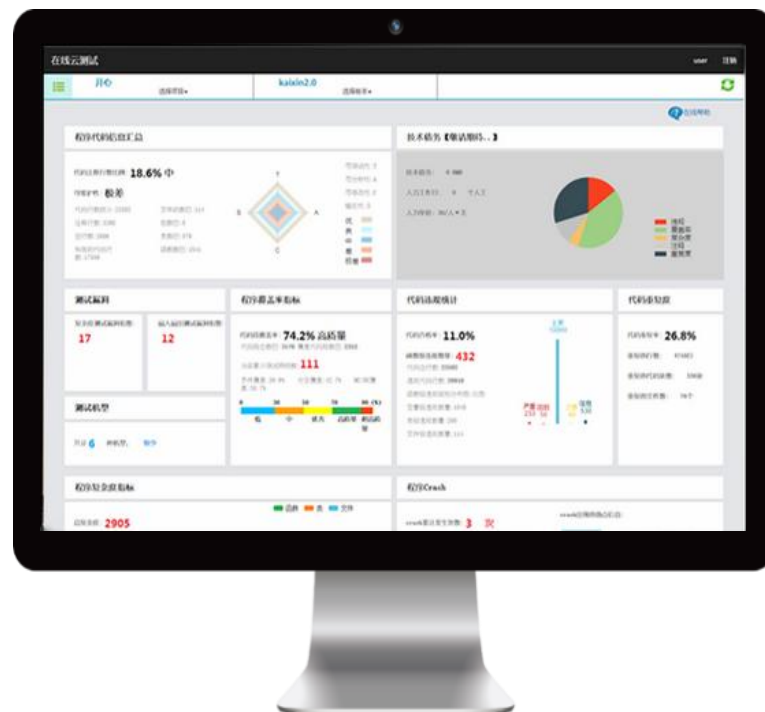
实现同一测试用例在多个程序版本下各种缺陷的追溯。



## 全方位的技术指标，提供数字化验收支持

技术债务分析模型，分析软件内部设计在未来维护中的附加成本。可以对软件交付和验收提供直观的参考。

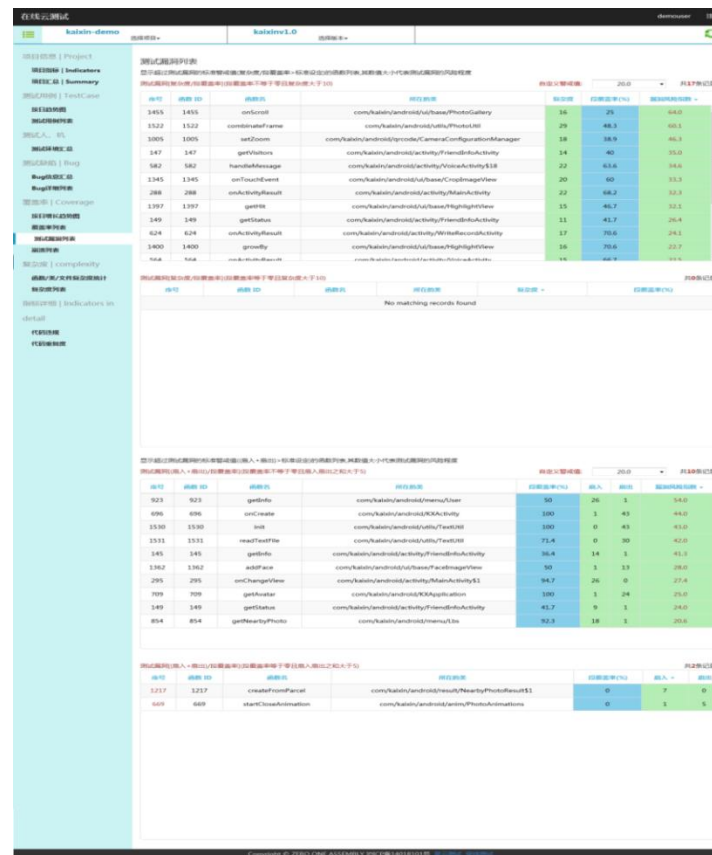
- 测试充分度
- 可维护性、可测试性
- 测试漏洞分析
- 代码重复块、代码违规、代码注释
- 测试贡献度分析、测试用例执行分析
- 崩溃的统计及代码级别崩溃分析



# 结合代码结构与动态数据的测试漏洞检出

通过静态、动态指标的综合分析，在大量的程序模块中，通过计算直接筛选潜在的高危的测试漏洞，可以在短期内确定高危漏洞模块，针对性的解决可以帮助用户快速找到严重缺陷。

- 通过复杂度和动态覆盖率进行计算。
- 通过函数调用上下文和动态覆盖率进行计算。



ID	Name	Status	Score	Weight	Score * Weight
1455	com.katun	通过	100	1	100
1522	com.katun	通过	100	1	100
1505	com.katun	通过	100	1	100
147	com.katun	通过	100	1	100
148	com.katun	通过	100	1	100
1345	com.katun	通过	100	1	100
288	com.katun	通过	100	1	100
1397	com.katun	通过	100	1	100
149	com.katun	通过	100	1	100
624	com.katun	通过	100	1	100
1400	com.katun	通过	100	1	100
504	com.katun	通过	100	1	100

### 智能回归测试用例选取



- 适应快速的版本迭代周期，适应庞大的工程项目。
- 在回归测试时，自动筛选测试用例，大大减少了回归测试的时间以及风险。

回归用例选取

回归生成测试用例数：117

排序：测试用例回归优先级

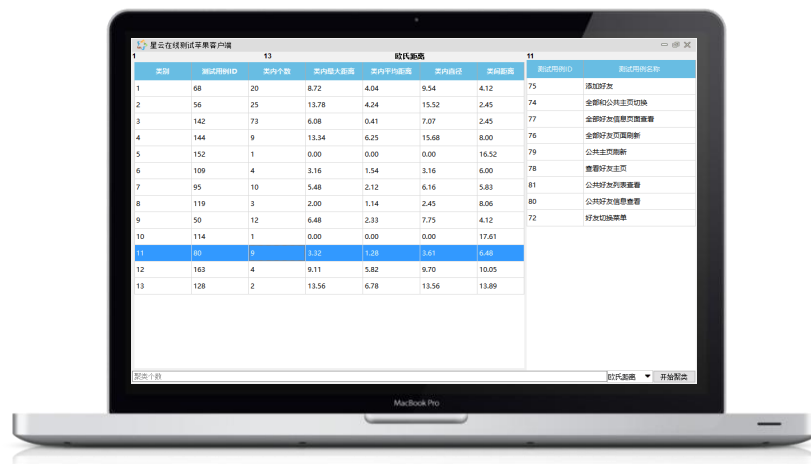
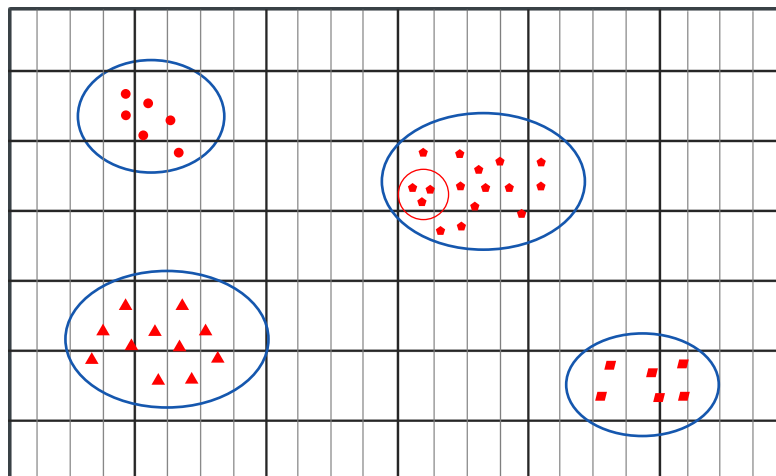
回归版本测试用例id	回归版本测试用例名称	测试用例最后运行版本	测试用例最后运行时间	测试用例最后运行状态	测试用例回归优先级	权重割面变化比例	
128	test	15/11/13	2015-11-16 14:02:49	BUG	120	29.4%	详细
13	附近酒店和菜单切换 (未登录)	15/11/13	2015-11-15 13:22:18	BUG	75	34.1%	详细
124	所有推荐酒店详细信息	15/11/13	2015-11-15 15:06:14	BUG	67	31.4%	详细
12	附近酒店和菜单切换 (登录状态)	15/11/13	2015-11-15 13:20:17	通过	63	29.2%	详细
127	测试补充	15/11/13	2015-11-16 12:21:13	通过	61	29.1%	详细
63	优惠券查看页面登录页面包围码	15/11/13	2015-11-16 11:30:50	BUG	56	30.6%	详细
20	附近酒店列表 (登录和未登录状态)	15/11/13	2015-11-15 13:48:53	BUG	48	37.9%	详细
45	酒店详细信息订单按钮	15/11/13	2015-11-15 14:26:08	BUG	47	29.9%	详细
90	删除我的收藏	15/11/13	2015-11-15 14:50:24	BUG	45	35.2%	详细
80	立即支付	15/11/13	2015-11-15 14:45:22	通过	45	41.1%	详细
84	删除我的订单列表	15/11/13	2015-11-15 14:46:37	BUG	43	41.7%	详细
11	附近酒店地图列表切换 (登录)	15/11/13	2015-11-15 13:18:49	通过	42	42%	详细
29	筛选	15/11/13	2015-11-15 14:07:49	通过	42	32.5%	详细
79	取消订单	15/11/13	2015-11-15 14:44:08	BUG	42	40.4%	详细
10	附近酒店地图列表切换 (未登录状态)	15/11/13	2015-11-15 13:13:39	通过	41	30.8%	详细
16	选择城市查看列表	15/11/13	2015-11-15 13:40:17	BUG	39	28.4%	详细

- 降低了传统人工回归分析产生的测试盲点。
- 精确计算回归用例的权重，测试人员在时间有限的情况下可以重点回归受改动影响最大的用例。



### 智能算法自动发现测试用例执行错误

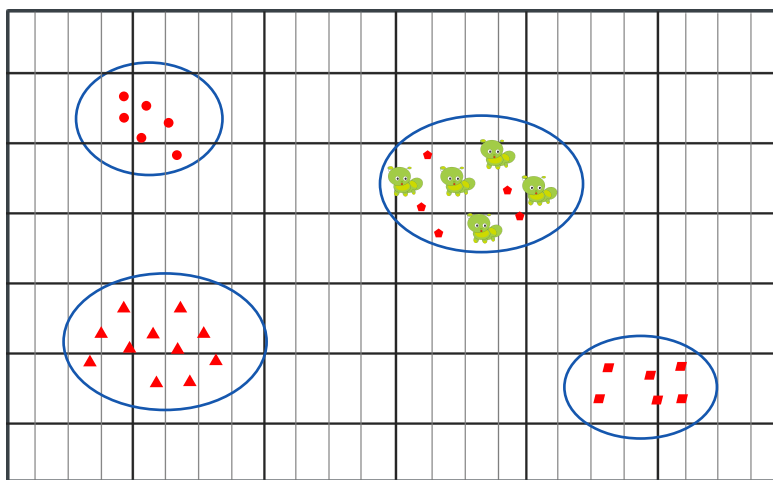
**大型系统维护**——根据测试用例的函数执行剖面的向量化信息，对测试用例进行聚类分析。



- 聚类结果可以分析被错误执行的用例，例如不相关的功能点聚类到一起，则说明其测试执行可能存在错误。

### 智能算法发现缺陷密集分布区

**大型系统维护**—根据测试用例的函数执行剖面的向量化信息，对测试用例进行聚类分析。



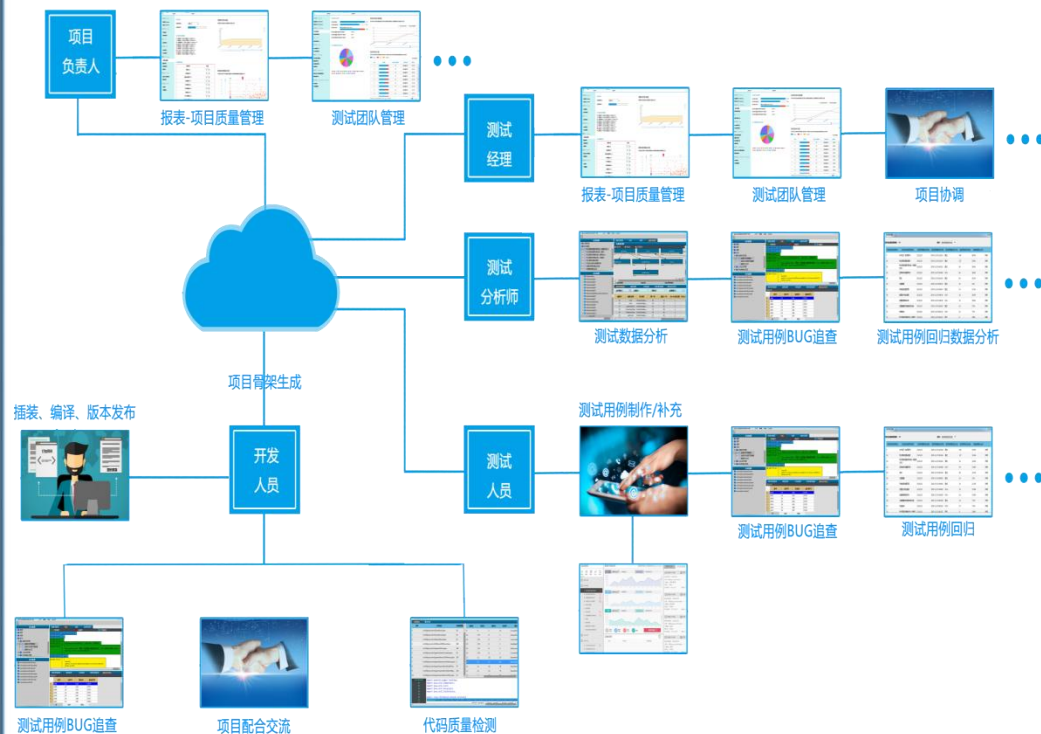
13 欧氏距离							11	
类别	测试用例ID	类内个数	类内最大距离	类内平均距离	类内直径	类间距离	测试用例ID	测试用例名称
1	68	20	8.72	4.04	9.54	4.12	75	添加好友
2	56	25	13.78	4.24	15.52	2.45	74	全部和公共主页切换
3	142	73	6.08	0.41	7.07	2.45	77	全部好友信息页面查看
4	144	9	13.34	6.25	15.68	8.00	76	全部好友页面刷新
5	152	1	0.00	0.00	0.00	16.52	79	公共主页刷新
6	109	4	3.16	1.54	3.16	6.00	78	查看好友主页
7	95	10	5.48	2.12	6.16	5.83	81	公共好友列表查看
8	119	3	2.00	1.14	2.45	8.06	80	公共好友信息查看
9	50	12	6.48	2.33	7.75	4.12	72	好友切换菜单
10	114	1	0.00	0.00	0.00	17.61		
11	80	9	3.32	1.28	3.61	6.48		
12	163	4	9.11	5.82	9.70	10.05		
13	128	2	13.56	6.78	13.56	13.89		

- 从类中检出中心点测试用例以及随机的其他用例，可以快速确定类中是否存在较多缺陷，快速定位缺陷的分布，并进行重点测试。



# 目录 在“云端”运行的白盒测试技术

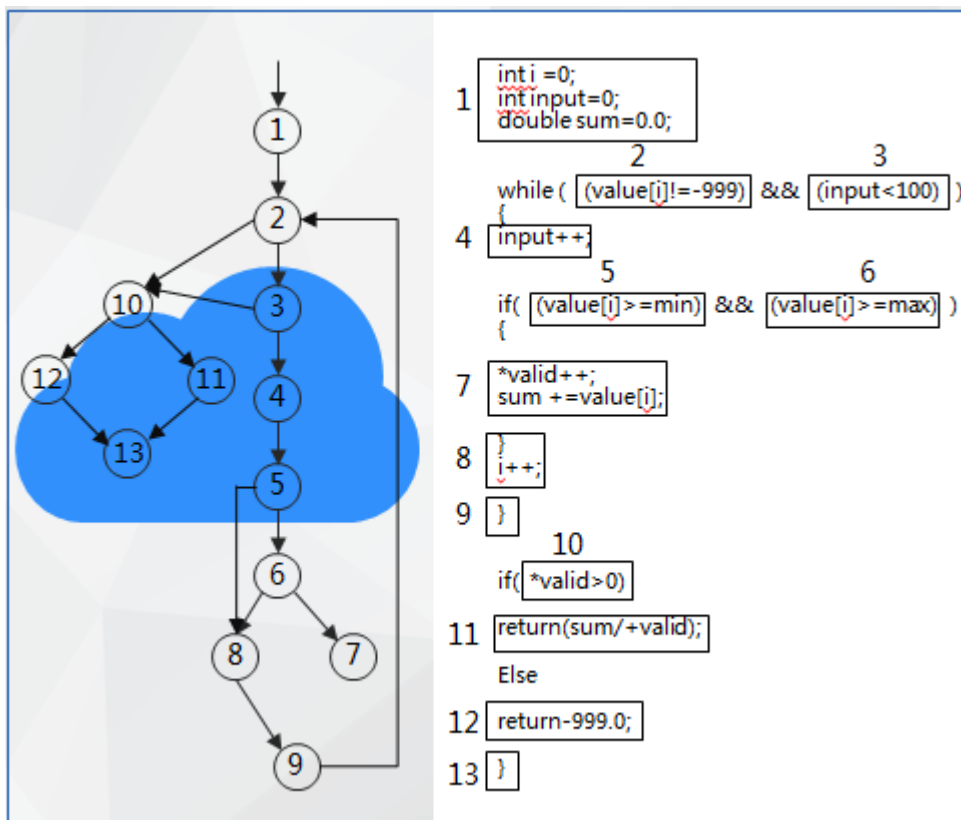
## 互联网测试执行流程：一键式透明运行方式



产品研发团队负责：创建项目、基于星云测试被测应用代码代码路径，一键式执行静态分析以及代码转换后并重新打包发布，分析互联网测试的结果

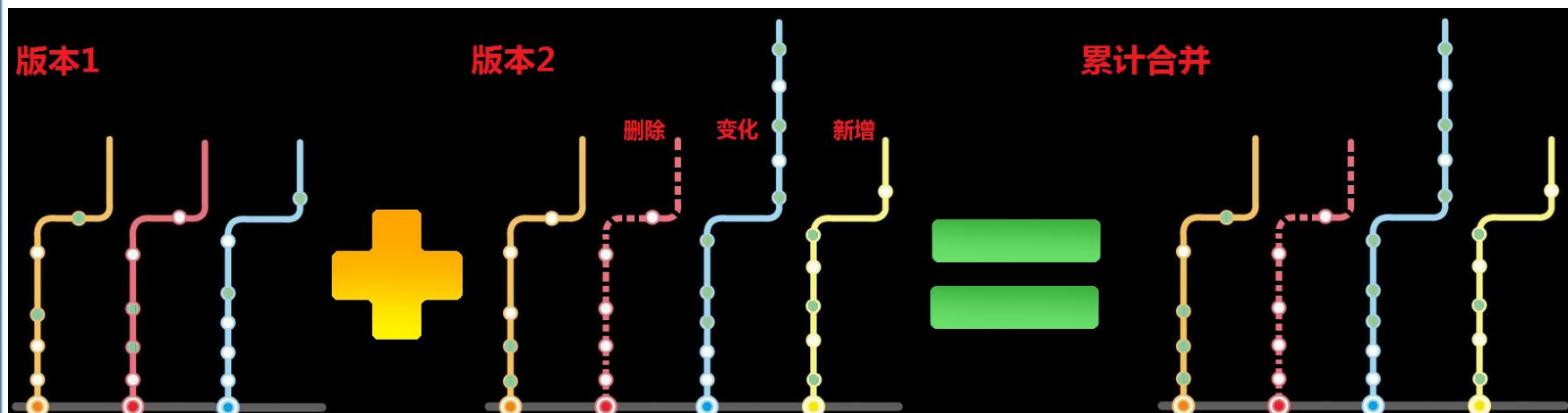
互联网测试团队负责：设计、执行黑盒测试用例。运行一遍，可实现秒速了解软件整体结构与布局、秒速了解测试用例、代码、模块的关联关系、秒速定位各种复杂BUG。与研发基于精准的测试数据进行工作协同。

## 云+客户端，无需上传源码的云端精准测试



- 企业私有云平台存储程序的“骨架”结构信息以及符号映射信息，可以通过云端符号信息与研发环境的代码随时进行关联。
- 测试结果均保存在云端服务器上，可以支持切换测试设备完成同步。
- 所有计算逻辑均在云端完成。

### 敏捷环境下多版本白盒测试数据的聚合



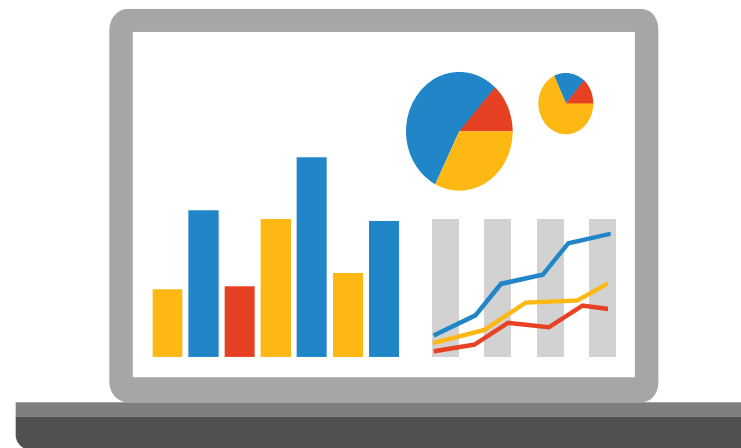
- 所有版本的测试数据，支持累计的合并计算，将多个敏捷局部测试的数据汇总到最新代码视图上统一展示。
- 可以任意选择版本合并，观察任何时间节点的累计数据。

### 技术特性

- 1、兼容被测应用语法规则，代码转换后不影响原有功能逻辑，额外性能开销小。
- 2、支持开放式的API方式（300+开放API），可以和自动化测试平台、测试管理系统以及各种云测试平台进行紧密的对接和整合，支持PAAS应用模式。
- 3、超高性能，每秒钟可处理10万余条程序运行、测试数据。
- 4、支持并发测试场景下，后台Web应用的数据路由和分发。



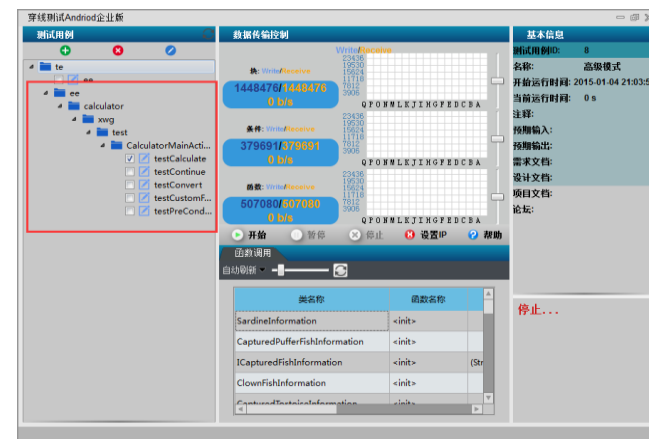
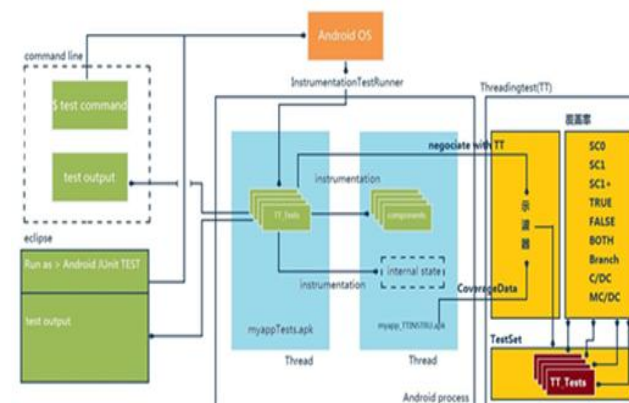
**星云测试的设计在可扩展性、高性能和平台扩展上进行了全面的考虑。**



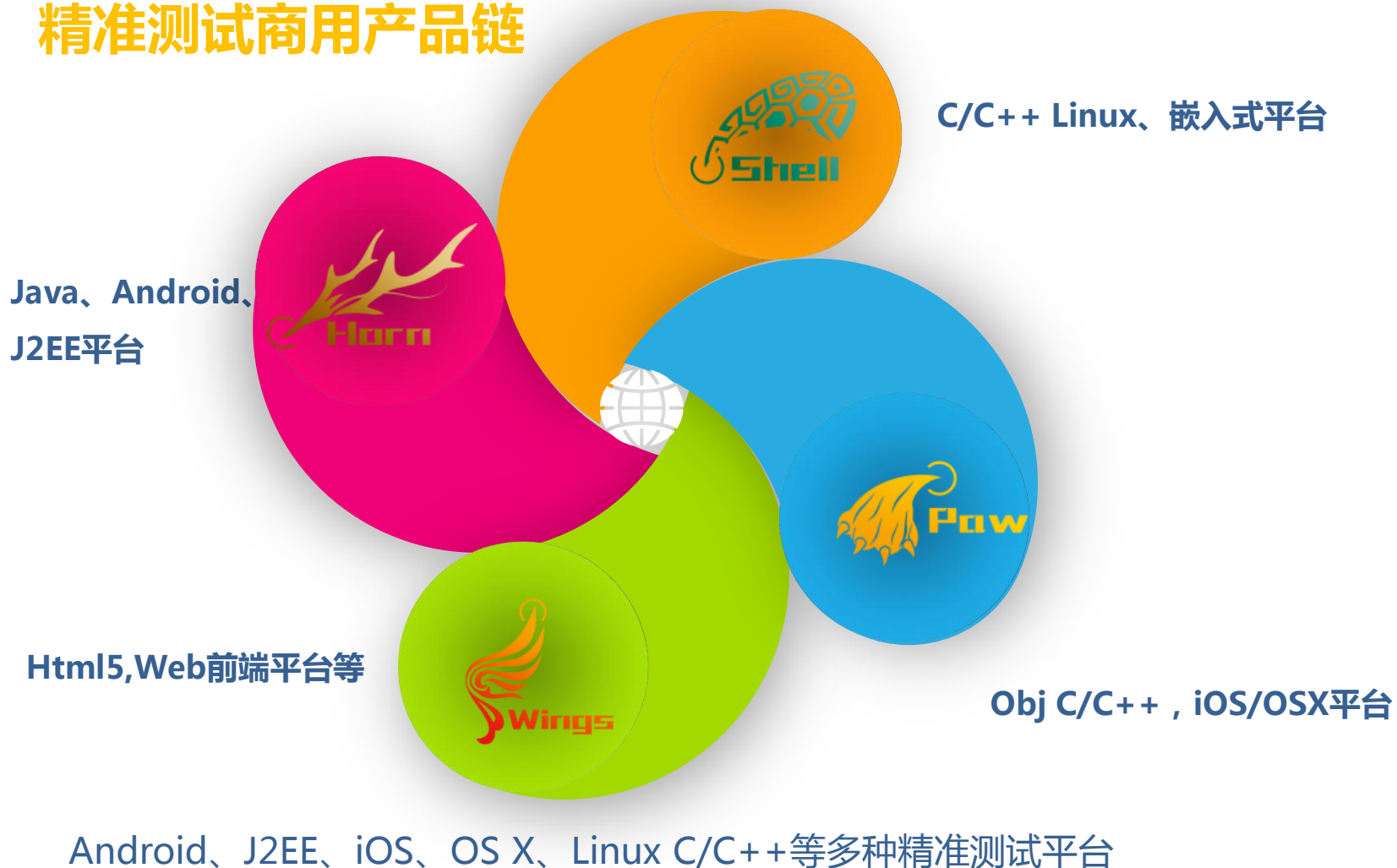
### 自由对接自动化测试

#### 精准测试（测试示波器）与自动化框架集成

- 示波器可以无缝的对接各种主流的自动化工具，测试人员无需改变原有的测试脚本流程，仅改变测试基类即可。
- 单元、自动化测试用例可以在产品中自动生成，无需事先手工建立。
- 全程无需人工干预。

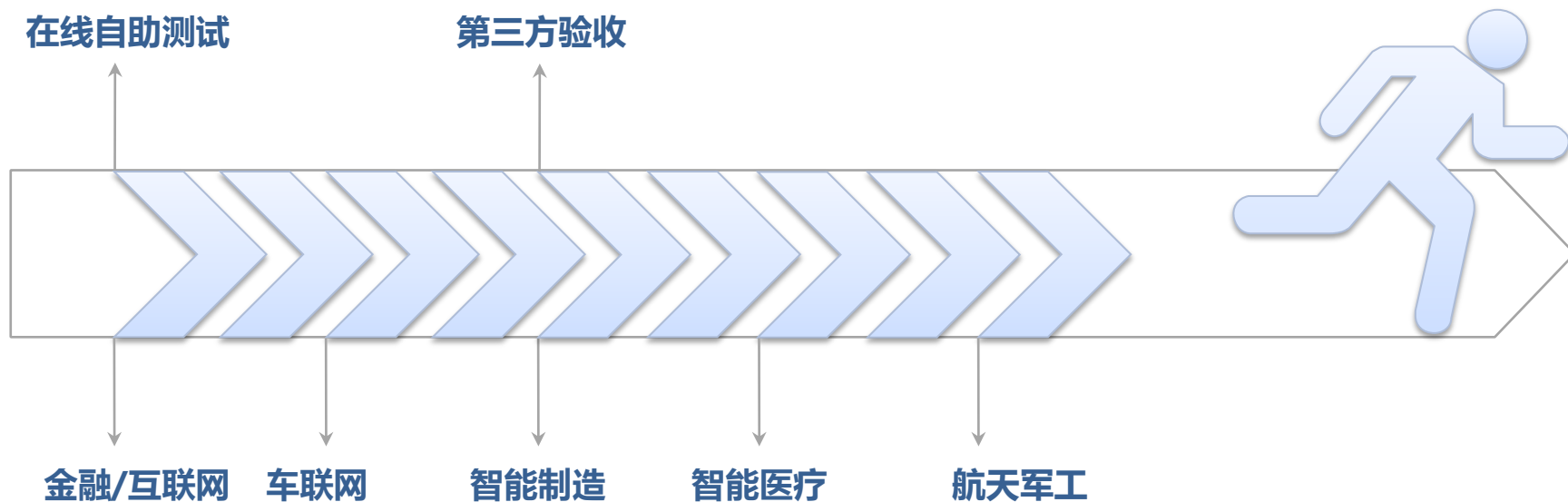


### 精准测试商用产品链



## 在线精准云测试的使用方式

### On Line -小客户+互联网测试模式



### Off Line -大客户

# THANKS





**Thank you**  
**ISTQB®让测试更专业**